Journal of Software Engineering and Applications, **2013, 6, 623-629**
Published Online December 2013 (http://www.scirp.org/journal/jsea)
http://dx.doi.org/10.4236/jsea.2013.612074

A Personalized Cloud Services Recommendation Based on Cooperative Relationship between Services

Chengwen Zhang[1], Jiali Bian[1], Bo Cheng[2], Lingfei Li[3]

[1]Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts & Telecommunications, Beijing, China; [2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing, China; [3]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT) of Ministry of Education, Beijing University of Posts & Telecommunications, Beijing, China.
Email: paperbupt@126.com

Received October 26th, 2013; revised November 20th, 2013; accepted November 27th, 2013

ABSTRACT

A personalized recommendation for cloud services, which is based on usage history and the cooperative relationship of cloud services, is presented. According to service groups, a service group could be defined as several services that were used together by one user at a time, and cooperative relationship between each two services can be calculated. In the process of recommendation, the services which are highly related to the service that the user has selected would be obtained firstly, the result should then take the QoS (Quality of Service) similarity between service's QoS and user's preference into account, so the final result combining the cooperative relationship and similarity will meet the functional needs of users and also meet the user's personalized non-functional requirements. The simulation proves that the algorithm works effectively.

Keywords: Personalized Recommendation; Cloud Service; Quality of Service; Similarity; Cooperative Relationship

1. Introduction

Recommendation system is an intelligent agent system to solve the information overload problem on the Internet, which recommends choices that meet users' needs or interests from a lot of information on the Internet to the users automatically. Personalized recommendation system is put forward as an independent concept in the 20th century until 1990, the first presented recommendation system is based on collaborative filtering recommendation [1], including content-based similarity [2] and user-based similarity [3] recommended methods. And in recent years label concept [4] has been presented to improve for user's single evaluation and the defects on the description of users and objects. Recommendation algorithm based on diffusion [5,6] and making use of graph theory bipartite graph [7-10] has also been presented, the physical diffusion theory is also used in the recommenddation system in order to improve accuracy and reduce complexity.

With the process of cloud computing in resent years, people begin to take their focus on the development of cloud service selection and recommendation [11-16]. Cloud computing environments utilize the SOA architecture, including three aspects, they are service providers, cloud providers and service users. Service providers no longer provide runtime environment for the release of the service, but utilize the cloud computing environment to run their specific services. Based on certain strategies, service providers select the appropriate cloud provider from the numerous cloud providers to run the specific services. And service users in the cloud computing environment select services which can meet their needs. The advantage of this architecture is that service providers and cloud providers run separately and have distinction permission, making logic service and the service environment separated, as well as bring about improving scalability and flexibility.

Cloud computing environment offers a wide variety of services that can be divided into different levels, from higher level to the lower level, they are Software application level services, platforms or environmental level services and infrastructure level services. Actually, the users usually choose a number of services spontaneously to form a combination to meet their own needs instead of using fixed combinations or a single service that cannot meet the needs of users. For example, a user wants to edit text online, in addition to using text editing functions (This feature can be seen as a software application level functions), it also needs to be uploaded to the Internet and can be downloaded in different places, it can be seen as a memory function in basic level of service. Cloud services are complementary to each other in functional aspects to achieve different goals. These services may include a variety of service units, such as hardware, software, platform, storage, computing, data and other services. According to different requirements and rules, services are combined to realize the complex functions.

According to the circumstances described above, a recommendation algorithm which is based on usage history and the combined relationship of cloud services is presented. Firstly, system will extract the users' usage of the group from services, then according to the group, calculate the cooperative relationship among services. While recommending, according to the service that user has chosen, system selects services which are highly related to the service that the user has chosen, and gives final result which combines the cooperative relationship and similarity between QoS and user's interests.

The remaining chapters are organized as follows: Section 2 describes some related works; Section 3 discusses in detail including model design and algorithm steps design; Section 4 presents the simulation work and results; Section 5 concludes the work and notes further research content.

2. Related Works

The most mature algorithm is the collaborative filtering recommendation algorithm [1], which includes user-based and content-based method. [2,3] discussed the two collaborative filtering algorithms in detail. With the development of the research on recommendation system, recommendation algorithm is no longer limited to collaborative filtering methods, but also make use of the network topology, this type of algorithms treats users and the projects as nodes, [5,6] discussed a network-based diffusion recommendation algorithm, according to the projects selected by user, it takes advantage of diffusion theory to find users who has similar interests to the target user and makes recommendation.

[8,9] presented an algorithm by making use of two sub-networks (also known as two parts diagram), according to the user's choice of products and creating the relationships through a diffusion algorithm, algorithm finds out users' potential demand or interest and makes recommendation. Algorithm presented in [9] weights edges in the network in order to improve the precision in recommending.

[7] presented a recommendation and clustering algorithm that works by creating a network topology based services relationship. Services are divided into two parts: services with large and small granularity, called SOS and S. SOS is usually composed by a variety of services S and some programming supplemented, the relationships of composition can be shown by a SOS-S network. According to this network structure, the diagrams that show the similarity relationship between large-grained services and the relationship between small-grained services can be calculated. Recommendation makes use of the combination relationship of small-grained services, and recommends the services which are closely related to according to the selected service.

3. Recommendation Models and Methods

Users and services are treated as nodes in this recommendation system, according to users' usage record, services will be combined to form a service group. Based on the group's composition and relationships, the cooperative relationship between services could be calculated, and system makes recommendation with the relationship. Compared to [7], service group is calculated and formed according to the usage of services, it is a logical concept to help analysis the relationship between services, but not a real entity existing in cloud. In the cloud environment, services that belong to different levels are often used together and form a service group, that brings about the situation that some lower-level services may become the fundamental member of service group, namely, this kind of service such as storage or platform is very necessary for most service group and can achieve a great many user's requires, and a higher-level service would be a professional characteristics of the group, this kind of service is just used in parts of service groups and is necessary for particular users in some professional fields. There are two kinds of relationships between services, which can be different for the user's current selection of services, and there are also large differences between recommending a base service when using a professional service and recommending a professional service when using a base service. So a two-way standard is introduced to distinguish the relationship of A to B and B to A when calculating the cooperative relationship between services. On the other hand, user's personalized interest should be taken into account while making recommendation.

3.1. Node Model

Users and services are treated as nodes in the system.

User often needs a set of services instead of a single one to complete a task within a certain period of time. In **Figure 1**, User U_1 selected services A, B and C to achieve his task, user U_2 selected services C, D and U_3 selected services C, E.

In users' history records, one user has chosen numbers of sets of service, so service group node was introduced to distinguish these sets and make it easy to analysis. In **Figure 2**, a service group includes some services, so the situation above could be described just like the structure in **Figure 2**, user U_1 used service group SC_1 and SC_2, and user U_2 used group SC_3.

User Model: User is defined as a set $U\{U_1,U_2,U_3,\cdots,U_i,\cdots\}$ which describes all users in the system. The user U_i has a set DS_i, defined as $DS_i=\langle U_i,SC,SC_m,SC_n,\cdots\rangle$, which describes all service groups that U_i had used. There is a vector QoS $\langle a,b,c,d\rangle$ for each user to describe the user's preference for stability, security, network quality and charge mode.

Service Group Model: Service group is defined as a set $SC\{SC_1,SC_2,SC_3,\cdots,SC_i,\cdots\}$, which describes all service groups in the system.

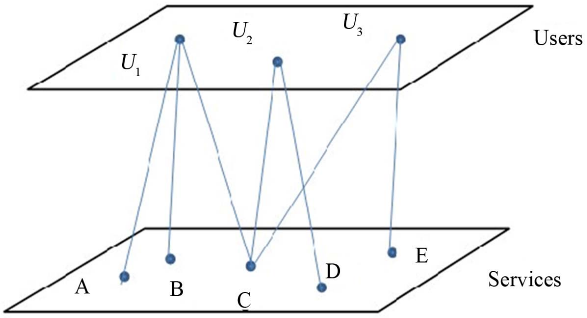

Figure 1. Structure of users and services.

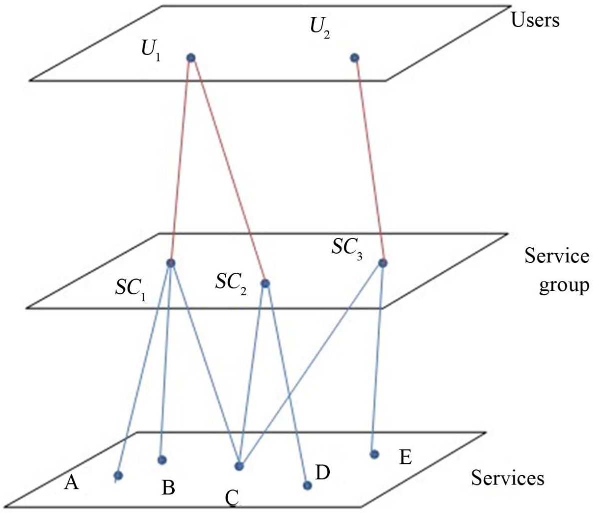

Figure 2. Model of service group.

Service Model: Service is defined as a set $S\{S_1,S_2,S_3,\cdots,S_i,\cdots\}$, which describes all services in the system. The service group SC_i, has a set D_i, which is defined as $D_i=\langle SC_i,S_j,S_k,\cdots\rangle$, describes the member services that make up the group SC_i. θ_{ki} is defined as the relationship of S_i and SC_k, it equals 1 when service S_i is one of members of service group SC_k, there will have an edge connected between the *i*-th service node and *k*-th service group node. N_k is defined to record the number of members of service group SC_k. φ_{ij} is defined to tell whether the *i*-th and *j*-th service are in the same service group or not. The relationship between services can be calculated only when they are in the same service group.

$$\theta_{ki}=\begin{cases}1,S_i\in D_k\\0,S_i\notin D_k\end{cases}\tag{1}$$

$$N_k=\sum_i\theta_{ki}\tag{2}$$

$$\varphi_{ij}=\begin{cases}1,\exists k,\theta_{ki}\cdot\theta_{kj}=1\\0,\forall k,\theta_{ki}\cdot\theta_{kj}=0\end{cases}\tag{3}$$

Cooperative Relationship between Services: Two services in one service group have two edges to connect. One is from *i* to *j*, another is from *j* to *i*. For the edge $S_i\to S_j$, there is a weight assigned to it, defined as ω_{ij}, means the cooperative relationship from S_i to S_j. ω_{ij} is defined as follow:

$$\omega_{ij}=\frac{\sum_k\theta_{ki}\theta_{kj}\dfrac{1}{N_k-1}}{\sum_k\theta_{ki}}\tag{4}$$

In the formula, $\theta_{ki}\theta_{kj}/(N_k-1)$ means the proportion for service S_j cooperates with S_i in all services that cooperate with S_i in the *k*-th group. ω_{ij} shows the probability of service S_j cooperates with S_i.

$$\sum_j\omega_{ij}=1\tag{5}$$

It is obviously that the relationship from S_i to S_j and the relationship from S_j to S_i are different. Since S_i and S_j may belong to different service grade or level, such as base services and professional services. When user selects a professional service, the recommendation system may give a base service with high probably, because the relationship from the professional service to the base service should be higher. Contrarily, when user selects a base service, the system might be confused to decide which professional service should be given because it does know user's profession, and the relationship from the base service to a professional service should be lower.

Similarity Between Users and Services: Vector representation is used to describe the QoS of services and users' preference, given two vector *x* and *y*, using cos(*x*,*y*) multiplied by mode of y to represent the similar *y* from *y* to *x*.

$$\mathrm{Sim}(x,y)=\cos(x,y)\cdot\|y\|=\frac{(x,y)}{\|x\|\cdot\|y\|}\cdot\|y\|=\frac{\sum_{i=1}^{n}x_i\cdot y_i}{\left(\sum_{i=1}^{n}x_i^{\,2}\right)^{1/2}} \tag{6}$$

Result of Recommendation: The final result should combine the cooperative relationship with QoS similar. So the result is defined as

$$R_{kij}=f\left(\omega_{ij}\right)+g\left(\mathrm{Sim}_{kj}\right)=a\omega_{ij}+b\mathrm{Sim}_{kj}.$$

where a = 20 and b = 0.004, a and b are set to standardize relationship and similarity.

3.2. Recommendation Method

The recommendation process is divided into four sections as followed: Section 1, data processing and creating set S and U; Section 2, getting users' preference; Section 3, obtaining users' usage and relationship of services; Section 4, service recommendation. Detail discussing follows:

1) According to records, the set of users U and the set of services S will be counted, and then the set of service group SC will be added up.

2) According to the evaluation of the users to the services, the vector of users' preference and Qos of services will be got.

3) The relationship between service groups and services and relationship between services will be calculated.

4) Recommendation. The input is user U_k and service S_i that the user has chosen. This process is divided into two steps; the first step is to calculate the top k services to which service S_i has higher cooperative relation, to meet the user's functional needs. This step must be done first, that means the service should match user's functional needs first. And then the second step is to calculate the QoS similarity relation between these *k* services and the user to meet user's non-functional needs. When the user has already selected more than one service, system should combine the result sets from each service according to the two steps above and give the final results.

4. Simulation and Analysis

4.1. Data Creating

Experimental data are created by a semi-random method, firstly, the services and their QoS are produced randomly; and users are produced as the same way. A vector is given to each user to show user's preference, which will be only used in the next step to produce service groups. The rule in the step of producing group sets is that, A user selects several services from the service sets, and system will calculate the similarity of service QoS and user's preference, then the top 2 - 4 services will be taken as a service group, the number of members of a group is got randomly from 2 to 4, and each user has a number of groups. Actually, when users select the services in actual life, there will be a number of services provided for the user, and user will select those which could fit user's preference, this step of producing service group above has just simulated the process of users' selection of services. The vector just given is only used in the step to produce groups, and the vector of users' preference in the recommendation process will be recalculated in recommendation steps later.

In the data producing process, the QoS of services that the user has chosen is similar to the user's preference, so the service which has poor QoS would be eliminated. Meanwhile, services which have similar QoS will be selected by users who have similar preference, in other words, users who have similar preference will choose same services, these services will be combined as a group for many times, and the relationship of these services will be relatively high. The argument above is just to proof that, in the data created, the relationships between services are not completely random, there are some services within which, every one of them has relatively high relationship to each other. So, the recommendation based on the date is meaningful.

4.2. Results and Analysis

The date have simulated a small cloud environment, data is created as follow: there are 100 users and every user has 100 service group records, each group has 2 to 4 members. In recommendation, the system will recommend to one user whom we call is the target user. To judge the recommendation is effective or not, the user's record will be divided into two parts, one is the experimental group, the other part is the comparison group. The experimental group as the usage history record to get the target user's preference. In each service group in the comparison parts, one or two services will be selected as the input, output is just the recommendation result, and it will be juxtaposed with the other services in this group. If one of the other services in the group is also a member of result, this case of recommendation will be treated as valid.

The simulation will be designed and analyzed on the bases of the number of total services, the proportion of the experimental group and the number of services that user has already selected. The result set will have five services, the simulation will carry out 100 users to make recommendation respectively, and system makes account of the average and max rate of the valid recommendation. The number of total services is set to 20, 30, 40, 50, the proportion of the experimental group is set to 20%, 40%, 60%, 80%, the number of services that user has already selected is set to 1, 2.

If the total number of services is set to 50; there will be 2500 relationships between services, and the max number of service groups in theory will reach $C_{50}^{2}+C_{50}^{3}+C_{50}^{4}$, about 250,000. There are 10,000 groups in the data, these groups are small parts of the number in theory, but it covers all the possibilities of all relationships. If the number of total services is set 40, 30 and 20, the max number of services will reach about 100,000, 30,000 and 6000.

Figure 3 shows the account of the average and max rate of the valid recommendation when the number of services that user has already selected is set to 1 and in **Figure 4**, the number is set to 2.

The line marked MAX represents the max rate of valid recommendation, and the one marked AVERAGE represents the average rate. Result shows that, the rate decreases with the increase of the number of total services, one reason is that the lager the number of total services is, the fewer chances that the groups in system can express the relationships between services, another reason is that the lager the number of total services is, the lower the percentage proportion the result set shares of all services and the lower proportion that the results can cover the selection of the target user.

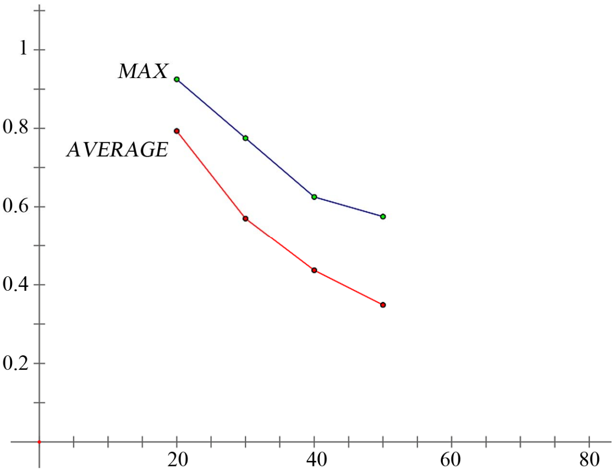

Figure 3. Results of rate when one service is selected.

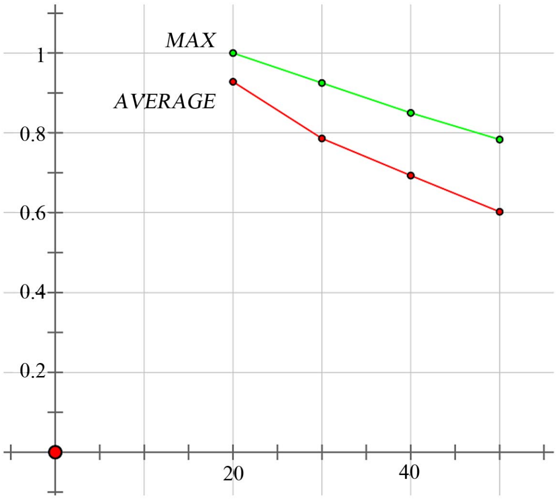

Figure 4. Results of rate when two services are selected.

In **Figure 5**, the number of services that user has already selected is set to 1, and the line above represents the average rate of valid recommendation when, the line below represents the rate in theory when the recommend to the target user randomly and with on recommend rules. When there are 20 services in system, the result set has 5 services, 25% of total services, but without recommend rules, system selects 5 services randomly, the proportion that the results can cover the target user's selection is about 43%, if the recommendation algorithm is used in the system, the coverage will reach 80%. It is believed that the recommendation algorithm is effective according to the two lines in **Figure 5**.

On the other hands, in the **Figure 6**, the #2 line represents the average rate of valid recommendation when the number of services that user has already selected is set to 2, and #1 line shows the rate when the number is 1. The more the services that user has already selected, the higher the rate is. The reason is that, when user has selected 2 services, the services that will be recommended must have high relationship with both of the two services, the range of selection will be decreased and the result will be more accurate.

To analysis the affection of user's preference to the result, the proportion of the experimental group is set to 20%, 40%, 60%, 80%, the number of total services is set to 50. In **Figure 7**, the lines from #1 to #4 represent the max and average rate of valid recommendation when the number of services that user has already selected is set to 2 and the max and average rate when it is set to 1. The larger proportion of the experimental group is, the more accurate result is. It is believed that considering users' preference and making the similarity of services' QoS and users' preference as one of bases of recommendation has just improved the accuracy of the recommendation

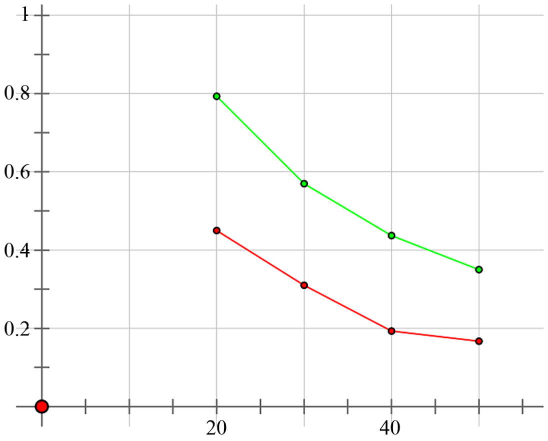

Figure 5. Recommendation and randomly selection results.

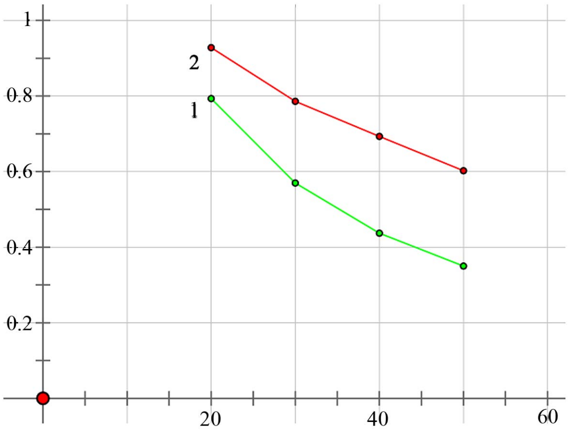

Figure 6. Average rate when one and two services are selected.

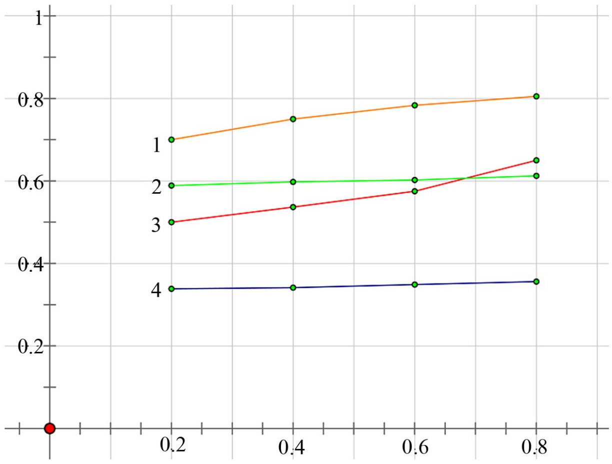

Figure 7. Results in different proportion of experimental group.

according to the #1 and #3 lines which represent the max rate. When the proportion is relatively small, the result is not too low. It is friendly to new users who do not have too many history records and system cannot get too much information about their preference.

5. Conclusions

Above all, a personalized recommendation for cloud services, which is based on usage history and the cooperative relationship of cloud services, is presented. Recommendation works according to the cooperative relationship among services to find out the functional complementarity among them, and to recommend the results. Performance in the simulation proves that the recommendation works well in a small-scale cloud environment with not so many services. One advantage of this recommendation is that, new users can get relatively effective recommendation without so much information about preference.

It remains future work to consider classification of users and services when the cloud environment scale is larger. Classification should be considered from the point of view to distinguish users' professionals and services' functions.

6. Acknowlegements

The work presented in this paper was supported by National Natural Science Foundation of China (Grant No. 61001118); Program for New Century Excellent Talents in University (Grant No. NCET-11-0592).

REFERENCES

[1] I. Cantador, M. Fe rnáandez and P. Castells, "A Collaborative Recommendation Framework for Ontology Evaluation and Reuse," Universidad Autóonoma de Madrid, Spain, 2006.

[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749.

[3] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems," Adaptive Web, 2007.

[4] Knowledge & Data Engineering Group (KDE), "Information Systems and Machine Learning Lab (ISMLL)," Tag Recommendations in Folksonomies, 2006.

[5] J. G. Liu, T. Zhou, Y. C. Zhang and Q. Guo, "Degree Correlation of Bipartite Network on Personalized Recommendation," *International Journal of Modern Physics C*, Vol. 21, No. 1, 2010, pp. 137-147. http://dx.doi.org/10.1142/S0129183110014999

[6] C.-X. Jia, R.-R. Liu, D. Sun and B.-H. Wang, "A New Weighting Method in Network-Based Recommendation," *Physics A*, Vol. 387, 2008, pp. 5887-5891.

[7] W. F. Pan, B. Li, B. Shao and P. He, "Service Classifica-

tion and Recommendation Based on Software Networks," *Chinese Journal of Computers*, Vol. 34, No. 12, 2011, pp. 2355-2369.

[8] T. Zhou, J. Ren, M. Medo and Y.-C. Zhang, "Bipartite Network Projection and Personal Recommendation," *Physical Review E*, Vol. 76, No. 4, 2007, Article ID: 046115.

[9] X. Pan, G. S. Deng and J.-G. Liu, "Weighted Bipartite Network and Personalized Recommendation," *Physics Procedia*, Vol. 3, No. 5, 2010, pp. 1867-1876. http://dx.doi.org/10.1016/j.phpro.2010.07.031

[10] Y. R. Gu and M. Chen, "One Tag Time-weighted Recommend Approach on Tripartite Graphs Networks," *Computer Science*, Vol. 39, No. 8, 2012, pp. 96-98.

[11] S. Kang, S. Kang and S. Hur, "A Design of the Conceptual Architecture for a Multitenant SaaS Application Platform," *Computers*, *Networks*, *Systems and Industrial Engineering* (*CNSI*), 2011 First ACIS/JNU International Conference on Digital 2011, pp. 462-467.

[12] K. S. Gopalan and S. Nathan, "A Cloud Based Service Architecture for Personalized Media Recommendations," *International Conference on Next Generation Mobile Applications*, *Services*, *and Technologies*, 2011, pp. 19-24.

[13] P. Bedi, H. Kaur and B. Gupta, "Trustworthy, Service Provider Selection in Cloud Computing Environment," *International Conference on Communication Systems and Network Technologies*, 2012, pp. 714-718.

[14] S. X. Yan, C. Q. Chen, G. P. Zhao and B. S. Lee, "Cloud Service Recommendation and Selection for Enterprises," 6*th International DMTF workshop on systems and Virtualization Management (SVM*2012*)/CNSM*, 2012, pp. 431-433.

[15] S. Bardhan and D. Milojicic, "A Mechanism to Measure Quality-of-Service in a Federated Cloud Environment," *Federated Clouds*'12: *Proceedings of the* 2012 *Workshop on Cloud Services*, 2012, pp. 19-24.

[16] W. Y. Zeng, Y. L. Zhao and J. W. Zeng, "Cloud Service and Service Selection Algorithm Research," *GEC*'09: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ACM, 2009, pp. 1045-4048.