W_SR: A QoS Based Ranking Approach for Cloud Computing Service

Arezoo Jahani, Leyli Mohammad Khanli, Seyed Naser Razavi

Department of Electrical and Computer Engineering, University of Tabriz, Iran
a.jahani91@ms.tabrizu.ac.ir, l-khanli@tabrizu.ac.ir, n.razavi@tabrizu.ac.ir

ABSTRACT

Cloud computing is a kind of computing model that promise accessing to information resources in request time and subscription basis. In this environment, there are different type of user's application with different requirements. In addition, there are different cloud Service providers which present spate services with various qualitative traits. Therefore determining the best cloud computing service for users with specific applications is a serious problem. Service ranking system compares the different services based on quality of services (QoS), in order to select the most appropriate service. In this paper, we propose a W_SR (Weight Service Rank) approach for cloud service ranking that uses from QoS features. Comprehensive experiments are conducted employing real-world QoS dataset, including more than 2500 web services over the world. The experimental results show that execution time of our approach is less than other approaches and it is more flexible and scalable than the others with increase in services or users.

Keywords: Cloud Computing, Cloud Service Provider, Quality of Service, Ranking.

1. INTRODUCTION

Cloud Computing is an internet-based model that provides three kind of services (software, platform and infrastructure [1, 2]). With the benefits of cloud computing also there are new challenges such as service selecting that should be truly addressed. Each user that wants to apply one cloud service for own application has different requirement [3]. Thus, selecting the best service that fulfills user's application requirements is an important research challenge [4-6].

For best service selecting, we need to compare services based on QoS information [7] that can be measured by service providers or by a third party [5] such as monitoring systems. The value of quality features represents degree of quality of services [8]. After service comparison, ranking can be a suitable alternative for let users to sight the results [6, 9]. In addition, users have essential and non-essential requirements. Essential are requirements that selected service should exhibit that. But non-essential requirements aren't more important in view of user [9]. This paper attempt to propose approach for finding top ranked services and rank them instead of all service ranking. This work can find the confine to make a response and accelerate service ranking [10]. On the other hand, at first we find some candid services that satisfy user's requirements and after that we find one decision number for each service that can use for ranking with sorting.

The rest of this paper is organized as follows: in section 2, related work are reviewed. Section 3 and 4, describes the SMICloud framework and quality features. Section 5 presents the W_SR approach which is evaluated in section 6. Section 7 concludes the paper with some future work.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2. RELATED WORK

This section reviews the approaches of service ranking such SMICloud(AHP base) [9], SVD base [11], Service Ranking System (SRS) [12], SLA(Service Level Agreement) Matching [13], CloudRank [14] and Aggregation [15] for cloud service ranking.

Kumar et al [16] proposed the SMICloud framework for service ranking based on analytical hierarchical process (AHP [17]). They use from CSMIC[1] standards which are proposed by CSMI consortium for extracting qualitative values that is require for service comparison and ranking. They use from monitoring tools [16] for obtain QoS features value. In cloud computing environment, monitoring tools are very useful like Cloudstone [18], CloudHarmony[19] and Cloud Sleuth[20] that are free and available on internet. Also CloudCmp[21, 22], YCSB benchmark [23] and CloudRank-D[24] are monitoring approaches in cloud computing environment.

Chan et al [11] proposed SVD based approach that uses Singular Value Decomposition (SVD) technique. This technique introduces a service mapper called cloud service provider mapper. This service mapper apply SVD technique on the Provider Quality(PQ) matrix(consisting of providers' information and qualitative features of providers) for extracting three singular value. Then, it can extract the most appropriate service among selected services by finding one pseudo-service.

Choudhury et al [12, 25] proposed a system called Service Ranking System (SRS). This system has two type ranking: static and dynamic. In the static ranking, all available cloud service providers are ranked without considering user requirements. But in dynamic ranking, suitable services ranked based on user requirements. Tejas et al proposed SLA matching approach [13] which define the process of identifying compatible cloud provider for a given requirements by matching SLA parameters [13]. Their work is a part of the Cirrocumulus project [26].

Zheng et al proposed CloudRank Approach [14] that is performed based on prediction of qualitative values. This work has mentioned that qualitative value of services require before service comparison which should be measurement. In traditional component based systems, invoke the components was used for measuring the values. But, it is impossible to apply invoke in cloud environment, because this task need a high time complexity and cost. Also, invoking usually would not achieve a correct answer due to Internet's unpredictable connections [14]. They import that invoking in cloud computing and each client-server environment can be performed in two different places. 1) Service provider side and 2) client side. In service provider side, very good values for qualitative measures will be achieved. Also, these values would be so close to the values that provider claimed to present. Against, if the invoking is performed in the user side, fewer values would be achieved in comparison to previous case. Because, Internet's connections are unpredictable and there is a geographical distance between users and providers. Also, different qualitative values would be received from each user. So, for achieving more realistic values, it is better to perform the invoking of service in client side.

Their approach exploited from personalized services in calling service model [27]. When a user requests for ranking, system gets the similarity of the user with systems previous users for which system has performed ranking. Then, the system applies the ranking presented for the previous user to the new one. Chunjie et al [24]

[1] Cloud Service Measurement Index Consortium

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

proposed Aggregation Approach that is based on the usage of benchmark tools and past users feedback information. They aggregate user's feedback(subjective assessment) and benchmarking result(objective assessment) for comparing and ranking of services [15, 24].

3. RANKING FRAMEWORK

We used the SMICloud [9] framework with a little conversion that proposed by kumar et al [9]. As depicted in figure 1 it has 4 main layers that contribute with each other to service ranking. Application layer receives user's application. SMICloud broker layer that is responsible for interaction with customers and understanding their application needs. Monitoring layer monitors the performance of cloud services. Service Catalogue layer has information about services and providers.

We changed the duty of two components: filtering and ranking. Filtering component uses for selecting candid services which can support user's requirement and ranking component have ability to give decision number to each candid service and rank them. Pursuant to figure 1, this framework has follow process for service ranking:

1) Service filtering component receive user's requirement in *Requirement* matrix.
2) Service catalogue component receive all service information (service quality).
3) Service filtering component, filters all services and find some candid services that satisfy user's requirement and send it to SMI calculator component.
4) SMI calculator sends candid services to Monitoring component to monitor them.
5) Service monitoring done.
6) Monitoring information send back to Monitoring component.
7) Information sends back to Ranking component.
8) Ranking component rank candid services and return the results to user.

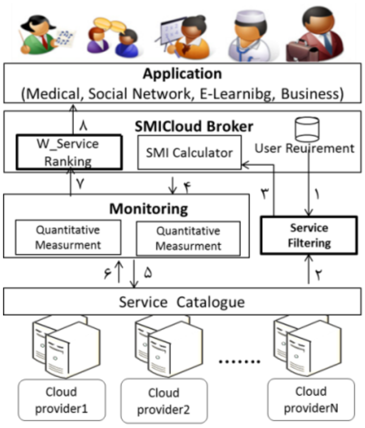

FIGURE 1 . Modified Ranking framework [9]

4. QUALITY FEATURES FOR SERVICE RANKING

We used from quality features that propose via CSMIC consortium [28] for cloud service comparison. These features are based on ISO and have 7 main features and many other sub features for each feature. Table 1 depicted CSMIC main features [9].

TABLE 1 .
CSMIC features

Feature	Definition
Acconutability	Possibility obtain to information and data in requirement time
Agility	Propagation or change service without any extra payment
Cost	Affectivity in cost
Performance	Such as service response time and accuracy
Assurance	likelihood of a Cloud service performing as promised in the SLA
Security	data protection and privacy
Usability	rapid adoption of Cloud services

5. PROPOSED APPROACH

As shown in figure 1, we had two changes in filtering component and ranking component. In proposed approach, we receive three inputs: User’s Requirement, Weights for each quality of service that display user’s priority for each quality of service and non-essential requirements. Figure 2 depicted how inputs use in proposed approach. Same as figure 2, Filtering component, receives three inputs and produce candid services and sends it to ranking component. After that ranking component, compare candid services and appointment one decision number for each candid service to Ranking. Rest of this section describes all details.

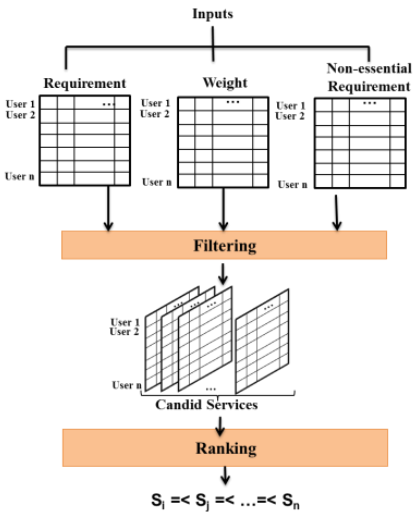

FIGURE 2. Inputs for filtering and ranking components

5.1 FILTERING PHASE

This phase done by filtering component and attempt to select k service as satisfy user’s requirement. In other word, this phase try to select k-top ranked services. Whereas user has essential and non-essential requirement, at first filtering component try to select services that satisfy both essential and non-essential requirement and if the size of candid services was lower than k, in second iteration omit one non-essential requirement with lowest efficacy Coefficient and try to select some other candid services and do this iteration to select all k candid services or there wasn’t any other non-essential requirement. This process depicted in figure 3 (Filtering box).

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

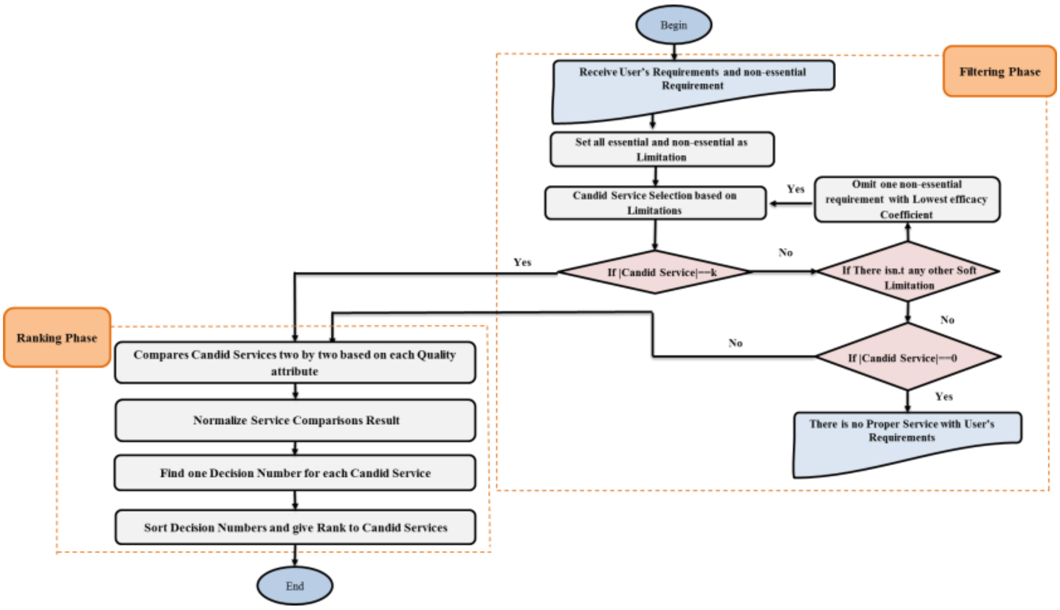

FIGURE 3. Flowchart of proposed approach

5.2 RANKING PHASE

This phase attempt to compare candid services based on each quality of service. For this aim, depict one matrix with candid services and quality's. This matrix, construct as figure 4 if assume there are k candid service (s_i, i=1,…, k) and Q quality of service (q_j, j=1,…, Q). Every entity in this matrix, as (cs_i, q_j) presents j^{th} quality of service in i^{th} candid service.

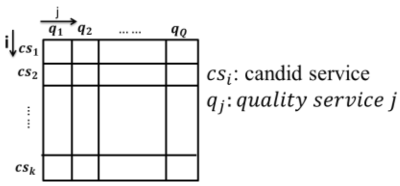

FIGURE 4 . Service comparison matrix

After construct service comparison matrix, because each quality of service has herself scale, we should normalize this matrix. With uses of algorithm 1, comparison matrix normalizes.

$$for\ j = 1\ to\ Q \qquad (1)$$

$$\text{Max}_j = \max\{q_j^{cs_{1,\dots,k}}\},\ \text{Min}_j = \min\{q_j^{cs_{1,\dots,k}}\};$$

for i= 1 to k

$$(cs_i, q_j) = (cs_i, q_j) - Min \,/\, Max - Min;$$

end for

end for

After normalize, Matrix entity's, was between 0 and 1. Now with use of algorithm 2, for each candid service, one decision number produces.

$$D(cs_i) = \min\{\max(\overline{W}_{ij}, (cs_i, q_j))\}; \quad (2)$$

$$\overline{W}_{ij} = 1 - W_{ij}, \quad i = 1,...,k, \; j=1,\dots,Q,$$

In algorithm 2, W_{ij} is weight that import by user and $D(cs_i)$ is decision number for i^{th} service. After procure decision number, service ranking done by sorting.

6. EXPERIMENT RESULTS

This section contain evaluation and experiments of proposed approach and describe the databese that uses for experiment results.

6.1 DATABASE DESCRIPTION

To evaluate W_SR approach accuracy, we conduct a large-scale real-world web service evaluation to collect QOS values on real-world web services with name QWS dataset including more than 2500 web services over the world.

6.2 PERFORMANCE COMPARISON

Our experiment implemented in MATLAB software. We have two experiments: flexibility and scalability. Flexibility test, represent our W_SR approach is more flexible than AHP approach thus W_SR have ability to increment each new service or new quality. Figure 5 depicted flexibility test via one abstract test.

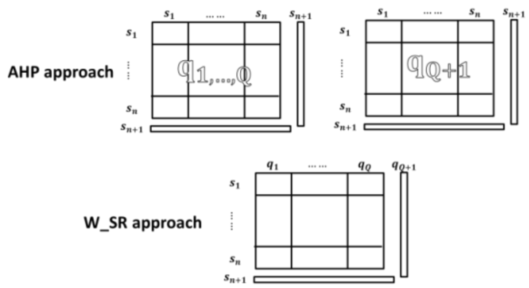

FIGURE 5. Flexibility test on AHP and W_SR approach

when we want to add one new service to AHP approach, the require time is same to adding one new row and one new column. But in W_SR approach the require time is same to adding only one row. About adding one new column, AHP approach add one new matrix with n+1 row and n+1 column (n is size of all services in dataset or catalogue). But W_SR add only one column that need to short time.

Scalability experiment has two types. First: with 2500 service and 100 candid service and 1 user. Two: with 2500 service and 200 candid service and 20 user. First schema depicted in figure 6(a) and second schema depicted in figure 6(b).

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

In figure 6(a), we changed number of services from 1 to 25000 with 1000 step. Tests results represent that with increase service number W_SR approach has linear increment and is more scalable than AHP. Although in low number of service SVD approach has low time consume but this approach can't be respondent in more than 11000 number of services. In result W_SR approach with one user is scalable than other approaches and need to low time to respondent.

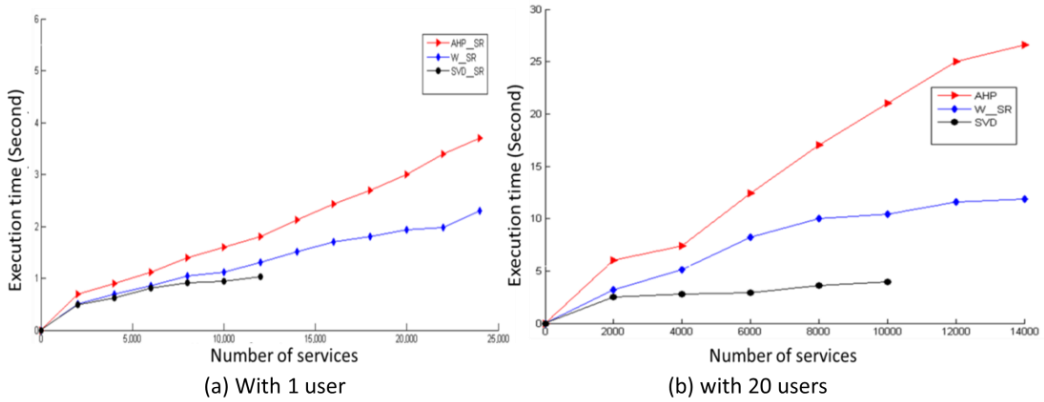

FIGURE 6. Comparison scalability with

In figure 6(b), we changed number of services from 1 to 14000 with 1000 step and premier than with 20 users. Test results represents that with increase service number, W_SR is scalable than AHP. Because AHP approach with increment each new service need to add one row and one column to her matrix to compare services and premier than AHP approach try to rank all services. But all service ranking isn't important. W_SR approach attempt to find response confine and select candid services and rank only candid services. Although SVD approach is scalable with low number of services, but it isn't able to answer with more than 10000 services when the number of users is 20.

As a result, we understand with these experiments that W_SR is more flexible and more scalable than the AHP and SVD approaches and can uses as bests service rank for cloud computing and can help to cloud users to evaluate services and select bests in low time.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose W_SR approach for cloud services, which requires to quality features and for service ranking receive three inputs. Our approach selects candid services that satisfy user's requirement. With this work, user doesn't amaze between lots of services and can simply find some services with himself requirements. We compare our approach with other two approaches and display that W_SR is more flexible and scalable than others.

For future work, we would like to improve the ranking accuracy and decrease the execution time of our approach by using train users experiments. Thus we want to use a machine learning to learn ranking results and use these result to answer to future users.

REFERENCES

[1] S. Ding, S. Yang, Y. Zhang, C. Liang and C. Xia, "Combining QoS Prediction and Customer Satisfaction Estimation to Solve Cloud Service Trustworthiness Evaluation Problems," *Knowledge-Based Systems*, Vol. 56, pp. 216-225, 2014.

[2] R. Buyya, C. Vecchiola and S.T. Selvi, "Cloud Computing Architecture," *chapter 4*, pp. 111-140, 2013.

[3] L.F.P. Eduardo Gonçalves, Marten van Sinderen, "Towards Runtime Discovery, Selection and Composition of Semantic Services," *Computer Communications*, Vol. 34, No. 2, pp. 159-168, 2011.

[4] M. Saravanan, "an Enhanced QOS Architecture Based Framework for Ranking of Cloud Services," *International Journal of Engineering Trends and Technology (IJETT)*, Vol. 4, No. 4, pp. 1022-1031, 2013.

[5] M.D. Stojanovic, S. Bostjancic Rakas and V. Acimovic-Raspopovic, "End-to-end Quality of Service Specification and Mapping: The Third Party Approach," *Computer Communications*, Vol. 33, No. 11, pp. 1354-1368, 2010.

[6] W. Itani, C. Ghali, A. Kayssi, A. Chehab, "Accountable Reputation Ranking Schems For Service Providers InCloud Computing," *International Conference on Cloud Computing and Services Science*, pp. 49-55, 2011.

[7] M. Katchabaw, H. Lutfiyya and M.A. Bauer, "Usage Based Service Differentiation for End-to-end Quality of Service Management," *Computer Communications*, Vol. 28, No. 18, pp. 2146-2159, 2005.

[8] V. Raisanen, "Service Quality Support—an Overview," *Computer Communications*, Vol. 27, No. 15, pp. 1539-1546, 2004.

[9] S.K. Garg, S. Versteeg and R. Buyya, "a Framework for Ranking of Cloud Computing Services," *Future Generation Computer Systems*, Vol. 29, No. 4, pp. 1012-1023, 2013.

[10] Y. Hao, Y. Zhang and J. Cao, "Web Services Discovery and Rank: an Information Retrieval Approach," *Future Generation Computer Systems*, Vol. 26, No. 8, pp. 1053-1062, 2010.

[11] H.C Chan, "Ranking and Mapping of Applications to Cloud Computing Services by SVD," *Network Operations and Management Symposium Workshops (NOMS Wksps)*, Osaka, 19-20 April, pp. 362-369, 2010.

[12] P. Choudhury, M. Sharma, K. Vikas, T. Pranshu and V. Satyanarayana, "Service Ranking Systems for Cloud Vendors," *Advanced Materials Research*, Vol. 433-440, pp. 3949-3953, 2012.

[13] S.C. Tejas Chauhan, Vikas Kumar and Bhise, "Service Level Agreement Parameter Matching in Cloud Computing," *World Congress on Information and Communication Technologies*, pp. 564-570, 2011.

[14] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, J. Wang, "QoS Ranking Prediction for Cloud Services," *Journal of IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 6, pp. 1213-1222, 2012.

[15] L. Qu, Y. Wang and M. Orgun, "Cloud Service Selection Based on the Aggregation of User Feedback and Quantitative Performance Assessment," *SCC '13 Proceedings of the IEEE International Conference on Services Computing*, IEEE Computer Society Washington, DC, USA, pp. 152-159, 2013.

[16] G. Katsaros, J. Subirats, J. Fito, J. Guitart, P. Gilet and D. Espling, "a Service Framework for Energy-aware Monitoring and VM Management in Clouds," *Future Generation Computer Systems*, Vol. 29, No. 8, pp. 2077-2091, 2013.

[17] O.S. Vaidya and S. Kumar, "Analytic Hierarchy Process: An Overview of Applications," *European Journal of Operational Research*, Vol. 169, No. 1, pp. 1-29, 2006.

[18] S. Will Sobel, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, Sh. Patil, A. Fox, D. Patterson, "Cloudstone: Multi-Platform, Multi-Language Benchmark and Measurement Tools for Web 2.0," *Provided by Sun Microsystems*, pp. 1-6, Oct 2008.

[19] "CloudHarmony," http://www.cloudharmony.com, 2009.

[20] T. Abubakr, "Cloud Sleuth," https://www.cloudsleuth.net, 2011.

[21] X. Ang, S. Kandula, M. Zhang, "CloudCmp: Comparing Public Cloud Providers," *ACM*, pp. 1-14, 2011.

[22] X. Ang, S. Kandula, M. Zhang, "CloudCmp Shopping for a Cloud Made Easy," *2nd USENIX Workshop on Hot Topics in Cloud Computing*, pp. 1-7 2010.

[23] B. Cooper, A. Silberstain, E. Tam, R. Sears, "Benchmarking Cloud Serving Systems with YCSB," *Proceeding of the 1st ACM Symposium on Cloud Computing, Indiana*, USA, 2010.

[24] J.Z. Chunjie, L. Wang, Gang LU, L. Zhang, C. Zhong, N. SUN, "CloudRank-D: Benchmarking and Ranking Cloud Computing Systems for Data Processing Applications," *Frontiers of Computer Science*, Vol. 6, No. 4, pp. 347-362,2012.

[25] S. Preeti Gulia, "Automatic Selection and Ranking of Cloud Providers using Service Level Agreements," *International Journal of Computer Applications*, Vol. 72, No. 11, pp. 45-52, 2013.

[26] "Cirrocumulus: A Semantic Framework for Application and core Services Portability Across Heterogeneous Clouds," *project at Kno-e-sis Center at Wright State University*, 2010.

[27] M.D. Dikaiakos and D. Zeinalipour-Yazti, "A Distributed Middleware Infrastructure for Personalized Services," *Computer Communications*, Vol. 27, No. 15, pp. 1464-1480, 2004.

[28] Cloud Service Measurement Index Consortium, "Service Measurement Index Version 1.0," *Carnegie Mellon University Silicon Valley Moffett Field*, CA USA, pp. 1-8, september 2011.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)