QoS based Enhanced Model for Ranking Cloud Service Providers

Narander Kumar[1] and Shalini Agarwal[2]

[1,2]Dept. of Computer Science
B. B. Ambedkar University (A Central University)
Lucknow, India

Abstract: Quality of Service provides an assurance that the requirements of users will be satisfied irrespective of system circumstances. Presently, there are no standards for cloud service provider interoperability, selection of software stack to meet QoS requirements and resource control and provisioning mechanisms. Cloud service providers often provide QoS through best-effort. However the Internet environment is unpredictable and therefore the perceived QoS by the users does not comply with the Service Level Agreement. Because several functionally equivalent cloud services are available, users need a decision making tool that can enable them to select the most suitable provider which can comply with their QoS parameters without fail. We propose a ranking model, which compares the service providers on various QoS criteria and ranks them according to their performances. We also perform a consistency evaluation of the service providers by comparing the entropy and hyper entropy of delivered QoS values to past users.

Keywords: Analytic hierarchy process; Consistency; Cloud service; Cloud service provider; Ranking; Quality of Service

I. Introduction

The Cloud Computing industry has showcased a wide variety of different types of services over the Internet. Many of these services have similar functions or utility. Therefore, when a user invokes a remote cloud service on the Internet, a plethora of alternatives from leading cloud companies are presented to him, each having a different price structure and performance level. In such cases, two types of decision situations are faced by the user. First, in which the user definitely knows the QoS criteria of the required service and also the importance of these criteria, no matter which alternative, in terms of service provider, the user may opt for. Such an understanding about one's own QoS criteria comes from experience but, the rapid growth in the number of cloud applications, requires him to update his judgment of the importance of the QoS criteria in view of the new knowledge.

Another type of decision situation is one in which the user is not sure of the QoS attributes of the service he wants to avail or application he wants deploy. For example, in selecting a storage server as a cloud service, the user may not know that whether security or availability is more important to him or by how much, unless the user actually tries some storage servers and experiences their performances. In this case, to have a better understanding of one's own QoS requirements the user needs to examine a wide selection of storage servers as they may be offering better reliability or elasticity or cost. Understanding of one's own or personalized QoS criteria and their relative importance are thus expanded.

Both types of decision situations present themselves frequently before cloud service users. An important point to note is that, in the first type of decision situation the user's QoS criteria decides which cloud service provider is right for the user. However in the second case, QoS criteria depend on the available list of cloud service providers. This type of situation is even more difficult to handle. Often the user may need to evaluate the tradeoff between the offered quality and the demanded cost. Therefore the selection of optimal cloud service provider is a complex task and an urgently required research problem.

This paper tackles the challenge of ranking cloud service providers according to QoS and presents an analytical model for Cloud Service Provider Selection Engine (CSPSE). The aim of CSPSE is to help the users to discover not only multiple cloud services and their providers but also to select the most suitable one in an objective way. QoS rankings are assigned to the cloud service providers so that a user can choose the required service easily. CSPSE model is designed in such a way so that it can address any number of users having a wide variety of QoS requirements. Also the number of QoS parameters can be extended to meet new and emerging requirements so that a high degree of personalization can be provided. User's personalized QoS criteria, their interrelationships and relative importances, for

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

judging the cloud service provider are modeled and stored using a quadtree structure so that the values can be quickly retrieved them for decision making.

With anything-as-a service on the horizon, the day is not far behind when common internet users would become service providers by dedicating their personal computers to others. This will transform service providing into a big commodity market and clearly indicates the important role of CSPSE model in the near future.

II. Related Work

The literature contains significant contributions in the area of quality of service (QoS) in distributed systems applications. However, QoS in cloud computing scenario is only scarcely been studied. In cloud computing scenario, QoS involves not only fine tuning of the system level algorithms but also necessitates the inclusion of certain techniques that give the users a quality of experience while using those services**(Adinolfi et al., 2012)**. **(Kourtesis et al., 2014)** explained the challenges faced in meeting QoS standards in cloud systems and presented a semantic based management framework which provides intelligent and interoperable environment for monitoring diverse services in clouds. In a similar work **(Kafetzakis et al., 2012)**, QoS in cloud perspective is measured in terms of the extent up to which the clauses in the service level agreement (SLA) are met. Violation of SLA lowers the customer's trust on the cloud service provider. **(Ding et al., 2014)** presented a scheme evaluating service trustworthiness based on predicted QoS and level of customer satisfaction. However using the scope of utilizing past users' experience of service satisfaction has not been explored as done in the present work.

Some researchers **(Yuchao et al., 2012)** presented the link between QoS delivery and service provider workload. They claim that in order to ensure the desired level of quality, a service provider must be moderately loaded so that it does not depart from the promised quality attributes such as response time and availability of resources, due to work overload. In this way QoS is also related to load balancing and therefore whenever applications are scaled up QoS must be maintained. However, comparison of service providers' workload and ranking according to least or most overloaded service provider, is not discussed. **(Alhamazani et al., 2012)** and **(Alhamazani et al., 2012)** suggested the use of automated quality monitoring systems to detect variations in service performance. But their works do not facilitate the user to select the service provider delivering least variation in service quality. In this paper, this challenge is tackled by using a method called inconsistency computation described in section.

Dynamic requests by cloud users lead to composition of services, a good account of which is given by **(Jula et al., 2014)**. However, simple and composite service selection from a pool of available services in the Internet is a NP-hard problem **(Zhao et al., 21012)** and several problem solving approaches are employed to get the solution for optimal selection of services. For fast and best selection of cloud services **(Zheng et al., 2013)** presented a technique of similarity computation that identifies past users with similar QoS requirements and then used the past choices to make current decisions. This approach may not be suitable in those cases where there is a large inconsistency in the performance of cloud service providers.

From the comprehensive literature study, it is observed although much work is devoted to the methods and mechanisms for enhancing QoS on the service providers' end but little consideration has been given to solving user's problem of discovering the best choice for his service requirements.

In an inspiring work of **(Garg et al., 2013)** an account of several metrics for the measurement of cloud service has been provided and cloud service providers are ranked on the basis of given metrics. However the possibility of variations in the metrics, that may alter the rank of a service provider, has not been explored.

In this paper we apply Analytic Hierarchy Process (AHP) **(Satty, 2008)**, a promising approach to solve Multi Criteria Decision Making (MCDM) Problem**,** for solving the ranking problem**.** AHP has a vast range of application in solving engineering problems as given by **(Lee et al., 2009)** and **(Triantaphyllou et al., 1995).** However, its use for QoS based ranking in cloud computing systems has not been much explored. We employ the cloud model given by **(Wang et al., 2011)** to analyze and enhance service characteristics for the proposed model. The proposed model is easy to implement and yields fast results as shown by our experiments.

In this paper the main contributions are-

- To present a model of cloud service provider selection engine (CSPSE) which is a decision making tool to help the cloud customers select the best

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

cloud service provider according to their personalized needs.

- To present a modeling technique for storing past users' priorities and choices.
- To present a technique of differentiating consistent and non consistent cloud service providers in terms of service delivery.
- To provide a QoS based ranking of cloud service providers.

III. Background and Motivation

There are several QoS parameters such as response time, cost and bandwidth, associated with a service. To compare and rank cloud service providers on a number of QoS parameters, we model the service provider selection problem as a MCDM problem. In the CSPSE model we use the Analytic Hierarchy Process, which is a promising approach for solving MCDM problems.

In AHP, complex problems are structured into a hierarchy of criteria, sub-criteria and decision alternatives from which the final choice is to be made. In contrast to a simple MCDM situation the criteria in AHP may be expressed in different units eg, time, currency, CO_2 emission, etc. and are often conflicting in nature. QoS criteria in Clouds have several dimensions and it is due to this reason AHP offers a great assistance in selecting an optimal service provider or the best service from among the given alternatives. To make decisions using AHP one identifies and analyses the tradeoffs between different alternatives to achieve an objective. A number of paired comparisons between the criteria and the alternatives are done iteratively. The final result is an ordered list alternatives according to user's preference. The structure of a typical AHP problem and its solution steps are described as follows.

We consider that the number of alternatives is P and the number of decision criteria is Q. Each of the P alternatives is evaluated in terms of each of the Q criteria to estimate the Relative Importance Value (RIV) of the P^{th} alternative in terms of Q^{th} decision criteria and is denoted by t_{ij} , where i = 1,2,3,.....P and j = 1,2,3,.....Q . Additionally, W_k denotes the weight of criteria Q_k. The MCDM problem can now be represented by a matrix as shown in Table 1. The values of the variables t_{ij} and W_k in table 1 are determined by performing pair-wise comparisons in three steps as described below-

Step 1- ***Compare alternatives in terms of decision criteria***

The alternatives (service providers in this paper) are compared to each other in terms of each decision criterion (QoS attribute in this paper) in a PxP matrix. The result is a vector denoting priority for each alternative in terms of each decision criterion. This vector is termed as a priority vector. The priority vector is obtained by computing right principal eigen vector of the PxP matrix (Satty, 1998). If the number of decision criteria is Q, then Q such pair wise comparison matrices are built.

Step 2 - ***Comparison among decision criteria***

The Q decision criteria are compared pair wise to obtain relative weights of importance in a QxQ matrix. The weights correspond to the right principal eigenvector of the QxQ comparison matrix.

Step 3 - ***Calculate the overall ranking***

Overall ranking of the alternatives (service providers) is calculated by using equation given below-

$$A^l = \sum_{k=1}^{Q} t_{lk} W_k, l = 1,2,3,..., P \quad ....(1)$$

TABLE 1. STRUCTURE OF AHP MATRIX

	Decision Ctriteria				
	C_1	C_2	C_3	...	C_Q
Alternatives	W_1	W_2	W_3	...	W_Q
A_1	t_{11}	t_{12}	t_{13}	...	t_{1Q}
A_2	t_{21}	t_{22}	t_{23}	...	t_{2Q}
A_3	t_{31}	t_{32}	t_{33}	...	t_{3Q}
.	.	.	.	.	.
.	.	.	.	.	.
A_P	t_{P1}	t_{P2}	t_{P3}	...	t_{PQ}

IV. Inconsistency Computation

The service providers are not able to provide the guaranteed QoS because several unforeseen circumstances existing in the Internet environment. In such a case the service provider selection engine should be able to select the provider which provides consistent QoS to the users. In other words, the offered QoS should not have a large variance. The process of evaluating inconsistency in the delivered QoS value is called as Inconsistency Computation. The process is explained with the help of an example below.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

Let CSP denote the output ordered list of all the service providers as derived from the AHP selection phase. We perform the test of consistency on the output list in the following way. Let C_1 be the QoS criterion having the largest weight W_j as judged by AHP. Further suppose that C_1 is negative attribute meaning thereby, high value of C_1 implies low quality, for example, Turnaround Time or Response Time. Let CSP_1 and CSP_2 are the two service providers which are to be judged for consistent performance on QoS criteria C_1. The performances of CSP_1 and CSP_2 are recorded by a series of transaction log entries which reflect the actual C_1 values delivered (and hence the fluctuations) by each provider during real time invocation. We illustrate five such log entries of CSP_1 and CSP_2 in Table 2.

TABLE 2. LOG ENTRIES OF ACTUAL QoS VALUES

CSP_1		CSP_2	
Transaction No.	**C_1 Value**	**Transaction No.**	**C_1 Value**
$CSP_1(1)$	39	$CSP_2(1)$	26
$CSP_1(2)$	36	$CSP_2(2)$	50
$CSP_1(3)$	40	$CSP_2(3)$	41
$CSP_1(4)$	36	$CSP_2(4)$	44
$CSP_1(5)$	36	$CSP_2(5)$	22
Average Value	37.4	**Average Value**	36.6

From the table it is clear that the average QoS value of C_1 given by CSP_1 is larger than that of CSP_2. Because C_1 is negative attribute, CSP_2 will obviously be preferred over CSP_1. However further analysis of individual transaction reveals that, in three transactions CSP_1 has provided a lower value of C_1 than CSP_2. This means that CSP_2 has more fluctuations on C_1 and therefore CSP_1 is more consistent than CSP_2.

In order to distinguish which cloud service provider gives a consistent performance on QoS criterion C_1 we adopt the Cloud Model given by (Wang et al., 2011). We define three numerical characteristics associated with QoS criterion C_1. These numerical characteristics are described below.

(i) Expected value of the QoS criterion C_i ($E[C_i]$)

The expected value of the QoS criterion Ci is given by (2)

$$E[C_i] = \frac{1}{n}\sum_{i=1}^{n} CSP(i) \quad ...(2)$$

where CSP(i) indicates transaction log entries of a CSP on QoS criterion C_i

(ii) Entropy of the QoS criterion C_i ($En[C_i]$)

The entropy of the QoS criterion C_i is given by (3)

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

$$En[C_i] = \frac{\sqrt{\pi/2}}{n}\sum_{i=1}^{n} \left| CSP(i) - E[C_i] \right| \quad ...(3)$$

(iii) Hyper Entropy of the QoS criterion C_i ($HEn[C_i]$)

The hyper entropy of the QoS criterion C_i is given by (4)

$$HEn[C_i] = \sqrt{V^2 - (En[C_i])^2} \quad ...(4)$$

where V is the variance given by (5)

$$V = \frac{1}{n-1}\sum_{i=1}^{n} (CSP(i) - E[C_i])^2 \quad ...(5)$$

The triplet ($E[C_i]$, $En[C_i]$, $HEn[C_i]$) is called the eigenvector of QoS criterion C_i. With the help of the eigenvector each cloud service provider can be distinguished from the other on a given QoS criterion or attribute. Using the above equations we compute the eigenvectors for service providers CSP_1 and CSP_2. The results are shown in the Table 3. From the table we observe that Entropy and Hyper Entropy of CSP_1 is less than CSP_2, therefore CSP_1 is a more consistent service provider that CSP_2.

TABLE 3. COMPARISON OF SERVICE PROVIDERS

	$E[C_1]$	$En[C_1]$	$HEn[C_1]$
CSP_1	37.4	2.105569	3.160364
CSP_2	36.6	12.63342	144.2508

From the table we observe that Entropy and Hyper Entropy of CSP_1 is less than CSP_2, therefore CSP_1 is a more consistent service provider that CSP_2. During actual deployment of the model the transaction log of past users can be utilized to obtain eigen vectors on each QoS attribute. The search space of functionally equivalent service providers can be reduced with this technique so that the 'winner' can be declared in minimum amount of time.

V. Architecture of cloud service provider selection engine

Figure 1 shows the architecture of Cloud Service Provider Selection Engine. The components of CSPSE are described below:

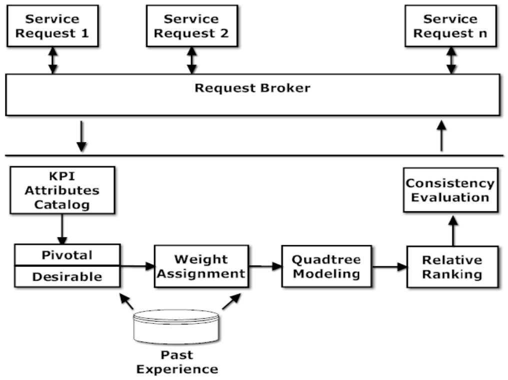

Fig. 1 Architecture of CSPSE

1. Request Broker- Users' requests for the ranking service originate dynamically with different QoS requirements. For example, a user deploying a network application essentially requires high throughput and low response time and cares little about the environmental impact or carbon footprints of the cloud service. The Request Broker is responsible for collecting personalized QoS requirements for various users. With the help of lower level components of the model, it can also provide suggestions to the users as to which quality attributes are crucial for the user's application and which ones are non-essential. This helps the naive users to better understand their needs by giving those examples of old users' having similar QoS requirements.

2. KPI Attribute Catalog- KPI or Key Performance Indicators are those quality attributes that judge the level of goodness of the cloud services. There is a vast range of KPI attributes mentioned in SMI. However all the attributes are not at the same level of importance for a given application. The KPI Attribute Catalog is a complete list of all such attributes or QoS parameters. Depending upon the QoS requirements collected by the Request Broker, the attributes in KPI Catalog are divided into two lists, viz. Pivotal and Desirables.

3. Pivotal vs. Desirables - The attributes recorded in the list Pivotal, are those QoS parameters that must necessarily be fulfilled by the service providers while deploying the user's application. While the pivotal attributes define the top level criteria, the entries in the list Desirables are those QoS attributes that may be compromised and define the lower level criteria according to which the service providers will be ranked. In order to help the cloud customers to clearly understand their application needs and accordingly help them to visualize the pivotal and desirable attributes, the CSPSE framework provides the past usage experience databases to enable the customers to take correct decisions.

4. Past Usage Experience Database- This database records old users' past experience of the perceived QoS offered by the service providers for the applications with similar quality requirements as the current user. This component helps to quickly filter the attributes of KPI Catalog into Pivotal and Desirables by extracting similar users from the database and employing their experience rating to assist the current user in performing an important task as described below. The AHP technique which we employ to give relative performance ranking to the contending cloud service providers requires user specified 'weight' or 'importance' to be given to all the QoS criteria in the top as well as lower levels. The PUED helps the new user to complete this important task.

5. Weight Assignment Component- This component assigns relative importance factor to all the QoS criteria included in the list Pivotal and Desirables. We explain the weight assignment process with the help of an example given below.
Example: Suppose the KPI attribute catalog has 8 attributes in all viz. K1, K2, K3, K4, K5, K6, K7 and K8. Let us assume that out of the 8 attributes 4 are Pivotal (say P1, P2, P3 and P4) while the remaining four are Desirable (say D1, D2, D3 and D4). Without loss of generality, we assume that, P1=K1, P2=K2, P3=K3, P4=K4 and D1=K5, D2=K6, D3=K7, D4= K8. Also it is assumed that D1 and D2 are the sub-criteria for P1, (For example if P1 denotes 'Agility', D1 may be 'Scalability' and D2 may be 'Transportability'. In other words Agility depends upon Scalability and Transportability). Similarly, D3 is the sub-criteria of P2 and D4 is the sub criteria of P3. P4 has no sub-criteria at all. Such type of interrelationship exists in practical scenario. Suppose that there are 4 cloud service providers (alternatives) named as CSP1, CSP2, CSP3 and CSP3 for a required service, our objective is to evaluate the relative ranking of the 4 pivotal (top level) attributes. Figure 2 shows the results.

Relative importance of top level attributes					
	P_1	P_2	P_3	P_4	Priority Vector
P_1	1	1/3	2	4	0.26
P_2	3	1	5	3	0.50
P_3	1/2	1/5	1	1/3	0.09
P_4	1/4	1/3	3	1	0.16

Fig. 2 Relative ranking of top level attributes

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

Next, we compute the relative ranking of the four cloud service providers with respect to P4. (Note that P4 has no further sub attributes). The results are shown in figure 3.

Relative Ranking of CSPs w.r.t. P_4					
	CSP_1	CSP_2	CSP_3	CSP_4	Priority Vector
CSP_1	1	5	3	7	0.55
CSP_2	1/5	1	1/3	5	0.13
CSP_3	1/3	3	1	6	0.27
CSP_4	1/7	1/5	1/6	1	0.04

Fig. 3 Relative ranking of service providers

In order to compute the Relative Ranking of CSPs with respect to P3, we first compute the Relative Ranking of CSPs with respect to D4 (the sub attributes) as explained above. We get the priority vector - (0.26, 0.14, 0.32, 0.47). Therefore, the relative ranking of CSPs with respect to P3 = (Relative Ranking of CSPs with respect to D4 * Relative importance of D4) i.e.

(0.26, 0.14, 0.32, 0.47) * 1 = (0.26, 0.14, 0.32, 0.47)

Similarly we compute that Relative Ranking of CSPs with respect to P2 as -

(Relative Ranking of CSPs with respect to D3 (the sub attribute) * Relative importance of D3) i.e.

(0.24, 0.25, 0.18, 0.33) * 1 = (0.24, 0.25, 0.18, 0.33)

To compute that relative ranking of CSPs with respect to P1 we first obtain the relative importance of the two sub attributes (desirables) D1 and D2 as shown in figure 4.

Relative Importance of sub attributes			
	D_1	D_2	Priority Vector
D_1	1	5	0.5
D_2	1/5	1	0.5

Fig. 4 Relative importance of sub attributes

Now the relative ranking of CSPs with respect to D1 is obtained as (0.15, 0.25, 0.13, 0.36) and the relative ranking of CSPs with respect to D2 is obtained as (0.22, 0.15, 0.11, 0.50).

Therefore the relative ranking of CSPs with respect to P1 is computed as –

$$\begin{pmatrix} 0.15 & 0.22 \\ 0.25 & 0.15 \\ 0.13 & 0.11 \\ 0.36 & 0.50 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 0.19 \\ 0.20 \\ 0.12 \\ 0.43 \end{pmatrix}$$

The final ranking of the CSPs with respect to all the attributes is now be obtained by using equation (1) given in Section III. The results are shown in figure 5.

	P_1 (0.26)	P_2 (0.50)	P_3 (0.09)	P_4 (0.16)	Final Ranking
CSP_1	0.19	0.24	0.26	0.55	0.2808
CSP_2	0.20	0.25	0.14	0.13	0.2104
CSP_3	0.12	0.18	0.32	0.27	0.1932
CSP_4	0.43	0.33	0.47	0.04	0.3255

Fig. 5 Final ranking of service providers

From the figure 5, we conclude that the ranking order of the four cloud service providers are as follows: CSP4, CSP1, CSP2, and CSP3. In this example we have considered four service providers and four pivotal attributes for the sake of simplicity. However any number of attributes and sub attributes can be taken to rank any number of CSPs by using similar computations.

6. Quadtree modeling – This component stores summarizes all the results obtained during the weight assignment phase in a quadtree data structure. Figure 6 gives the quadtree representation of the example discussed above. Because we considered four pivotal attributes, we use a quadtree for the summarized representation. However octree or some other larger representation can also be used. At the top level of the quadtree there are four cells that show the relative importance (weights) of the pivotal attributes. Each cell points to four more cells which either show the weight of the sub attributes or are kept null if there are no sub attributes.

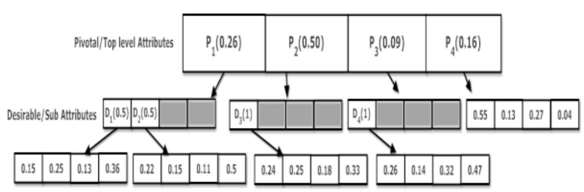

Fig. 6 Quadtree Modeling

7. Relative Ranking- This component computes the final relative ranking of the cloud service providers according to the values inserted in the quadtree. It then provides the ranking order to the next component.

8. Inconsistency Computation - The ranked list of the CSPs contains all the CSPs in the order of their service quality. However, they may not be consistent in providing the promised QoS to the users due to a number of unprecedented situations. This component

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

extracts the top ‘k’ service providers from the list and performs their consistency evaluation by comparing their eigenvectors as described in Section III (B). Thus the top ‘k’ providers are then re-ordered from the most consistent to the least consistent and the final result (top ‘k’ providers) is reported to the request broker which in turn displays it to the service user.

VI. Experiments and results

(a) In our study, the first set of experiments is conducted on the numerical example given in Weight Assignment Component in section IV. The four pivotal QoS attributes are taken as Response Time, Elasticity, Cost and Availability in the given order. The radar graphs given in figures 7 to 10, show the performance of four cloud service providers with respect to all the QoS attributes. Figure 10 shows that although CSP_1 is best in regard to response time, but it also incurs high cost for its customers. Also, because of the poor availability, it suits better for complex scientific applications rather than business purposes.

(b) The second experiment is conducted using a real-world QoS dataset WS-DREAM available as a web service. The dataset contains more than 1 million records of web service invocations of 150 users across 24 countries on 10,258 web services. In this experiment, we use our CSPSE framework to extract top ten service providers. After that, consistency evaluation of the values of the Response Time of these service invocations is done. Fig. 11 shows consistency level of the best (first) service provider is lower than that of the third service provider. This means that even though the first service provider has the minimum response time, but there are likely to be more fluctuations in the promised value. Therefore the third candidate with the most consistent delivery may be preferred.

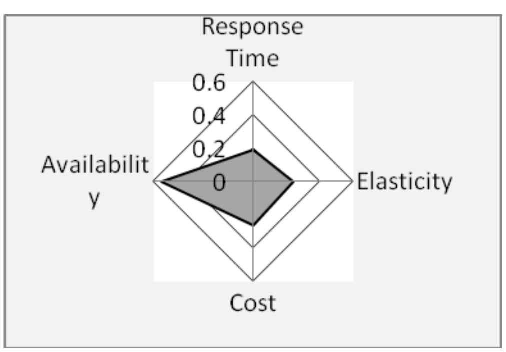

Fig. 7 Performance of CSP_1

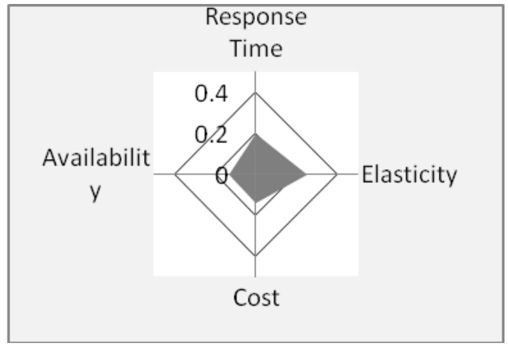

Fig. 8 Performance of CSP_2

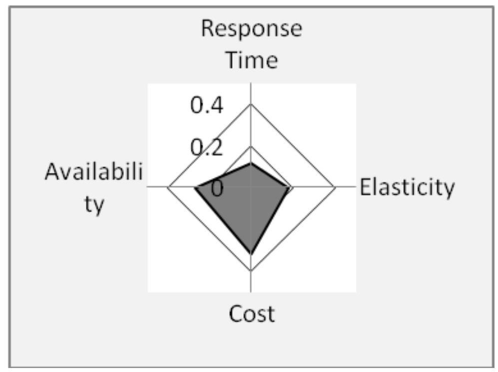

Fig. 9 Performance of CSP_3

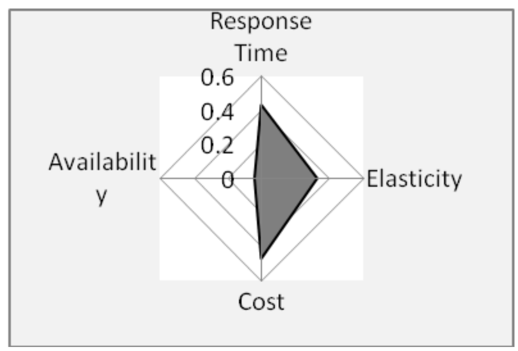

Fig. 10 Performance of CSP_4

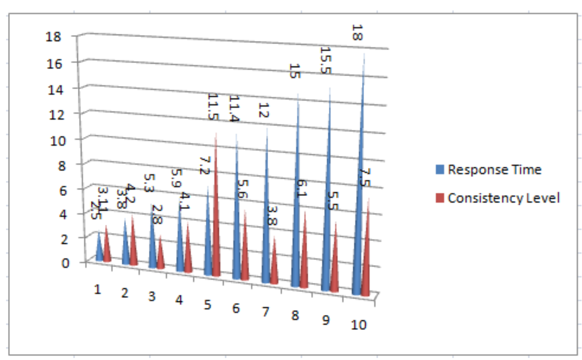

Fig. 11 Consistency Evaluation Graph of Top Ten Service providers

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)

VII. Conclusion and future work

With the exponential increase in the number of cloud service providers in IT landscape, it has become extremely difficult for the cloud users to know the best cloud provider that suits an individual's quality requirements. Sometimes the user is unable to understand which quality attributes are vital for his application and on what basis one should rank the available service providers. The proposed CSPSE framework not only helps the service users to understand one's own need but also serves to assign a ranking order to the cloud service providers on that QoS attributes which are vital for a specific user. The framework is novel in its approach because it also includes consistency evaluation of the cloud service providers which helps to reduce the search space as well as select that provider which always delivers the promised values of QoS parameters. Hence the framework is useful for cloud service users for service selection. It can also be used by a service provider for comparing its own performance with the other providers in the market and hence can be used as a monitoring tool.

As a future work we will implement the model using other techniques of multicriteria decision making such as mixed integer and fuzzy programming.

References

[1] Adinolfi, O., Cristaldi, R., Coppolino, L. and Romano, L., 'QoS-MONaaS: A Portable Architecture for QoS Monitoring in the Cloud', Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), **Vol. 527 No. 532, pp. 25-29, (2012)**

[2] Alhamazani, K., Ranjan, R., Rabhi, F., Wang, L. and Mitra, K., 'Cloud monitoring for optimizing the QoS of hosted applications', IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), **Vol. 765 No. 770, pp. 3-6, (2012)**

[3] Ding, S., Yang, S., Zhang, Y., Liang, C. and Xia, C., 'Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems', Knowledge Based System, **Vol. 56, pp. 216-225, (2014)**

[4] Garg, S. K., Versteeg, S. and Buyya, R., 'A framework for ranking of cloud computing services', Future Generation Computer Systems, **Vol. 29 No. 4, pp. 1012-1023, (2013)**

[5] http://www.wsdream.net

[6] Jula, A., Sundararajan, E. and Othman, Z. (2014), 'Cloud computing service composition: A systematic literature review', Expert Systems with Applications, Vol. 41 No. 8, pp. 3809-3824.

[7] Kafetzakis, E., Koumaras, H., Kourtis, M.A. and Koumaras, V., 'QoE4CLOUD: A QoE-driven multidimensional framework for cloud environments', International Conference on Telecommunications and Multimedia (TEMU), **pp. 77-82, (2012)**

[8] Kourtesis, D., Alvarez-Rodríguez, J.M. and Paraskakis, I., 'Semantic-based QoS management in cloud systems: Current status and future challenges', Future Generation Computer Systems, **Vol. 32, pp. 307-323, (2014)**

[9] Lee, H., Lee, S. and Park, Y., 'Selection of technology acquisition mode using the analytic network process', Mathematical and Computer Modelling, **Vol. 49 No. 5-6, pp. 1274-1282, (2009)**.

[10] Saaty, T.L., 'Decision making, new information, ranking and structure', Mathematical Modelling, **Vol. 8, pp. 125-132, (1987)**

[11] Saaty, T.L., 'Decision making with the analytic hierarchy process", International Jornal of Services Sciences, **Vol. 1 No. 1, (2008)**

[12] Saaty, T.L. , Hu G., 'Ranking by Eigenvctor versus other methods in the Analytic Hierarchy Process', Appl. Math. Lett. **Vol. 11, No. 4, pp. 121-125, (1998)**

[13] Triantaphyllou, E. and Stuart H. Mann, S.H., 'Using the analytic hierarchy process for decision making in engineering applications: some challenges', International Journal of Industrial Engineering: Applications and Practice, **Vol. 2 No. 1, pp. 35-44, (1995)**

[14] Wang, S., Zheng, Z., Sun, O., Zou, H. and Yang, F., 'Cloud model for service selection', IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), **Vol. 666 No. 671, pp. 10-15, (2011)**

[15] Yang, X., Yan, L. and Zeng, L., 'How to handle uncertainties in AHP: The Cloud Delphi hierarchical analysis', Information. Sciences, **Vol. 222, pp. 384-404, (2013)**

[16] Yuchao, Z., Bo, D. and Fuyang, P, 'An adaptive Qos-aware cloud', International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM), **Vol. 160 No. 163, pp. 8-10, . (2012)**

[17] Zhao, L., Ren, Y., Li, M. and Sakurai, K., 'Flexible service selection with user-specific QoS support in service-oriented architecture', J. Network and Computer Applications, **Vol. 35 No. 3, pp. 962-973, (2012)**

[18] Zheng, Z., Wu, Z., Zhang, Y., Lyu, M.R. and Wang, J., 'QoS Ranking Prediction for Cloud Services', IEEE Transactions on Parallel and Distributed Systems, **Vol. 24 No. 6, pp. 1213-1222, (2013)**

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.4)