Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Dynamic Ranking and Selection of Cloud Providers Using Service Level Agreements

Preeti Gulia, Sumedha Sood
Department of Computer Science and Applications
M.D. University, Rohtak, Haryana, India

Abstract— Cloud computing is gaining huge attention these days. Service level agreements play a crucial role in cloud computing. A provider offering good SLA will always have an edge over other providers in market as consumers would prefer the cloud that offers good SLA as compared to others. Service level agreements help a cloud consumer to choose the best cloud by matching his requirements with the services offered by clouds as mentioned in their service level agreement. However, this process of matching becomes quite complicated when performed manually by a cloud customer. The reason for the same can be attributed to increasing number of providers offering services with growing popularity of cloud computing. As a result a consumer may make a wrong decision and thus risk his entire business on an unsuitable cloud. Hence, keeping the above issue in mind we propose a framework that removes the burden of selection from the customer by automatically selecting the best cloud for a user. The given approach selects the most suitable cloud by comparing different clouds on the basis of service they offer and on the basis of user requirements and priorities.

Keywords— Cloud Computing, Service level agreement, cloud provider, business level policies, performance objectives.

I. INTRODUCTION

As the world is advancing, the role of IT in our daily lives is increasing rapidly. As a result, the demand of computing power is also increasing day by day. While there was always an increasing demand for high processing power and resources at the industry level, computing needs at the personal level are also increasing at a very high rate [1]. All these factors have lead to success of cloud computing. This is so because with cloud computing one does not need to purchase computing power and resources but can easily access them on the rental basis (pay as you use basis) from various organizations (providers). Cloud computing can be defined as a novel computing paradigm in which various computing resources such as hardware and software that are available in remote locations can be accessed according to pay as you use model by means of internet. [2]Since cloud computing removes the overhead of buying entire computing power and resources, it proves to be much more cost effective than previous traditional approaches. There is also elimination of maintenance costs with cloud computing and various additional features such as scale up and scale down of resources are also available.

Service level agreement play an important role in cloud computing. They can be defined as short documents that contain various performance promises made by a provider. They also contain penalties that a provider would have to pay in case he is not able to deliver promised services. [3] While a provider can use SLA to promote his services [4], a consumer can use the same to select the best cloud by matching SLAs offered by different clouds with his own requirements.

Following are the some important points that must be considered while evaluating a cloud service level agreement [5]:

1. Understand Roles and Responsibilities: The cloud consumers should be clear about their roles and responsibilities which are contained in SLA. For this they should be able to identify each and every actor involved in cloud computing environment. The cloud consumer should also be able to identify activities of actors and clearly define their requirements and desired service levels.
2. Evaluate business level policies: The business strategies and policies developed for the business purpose and the policies contained in service level agreement are interdependent on each other. Hence, the cloud customer should carefully evaluate all the data policies as well as business level policies of the cloud provider as all these policies directly affect cloud consumer's cloud strategies as well as business cases.
3. Understand service and deployment differences: The cloud services offered mostly fall in one of three main categories namely IaaS, SaaS or PaaS. The SLAs for all these categories are different from each other. This is so because service level objectives, key performance indicators and level of cloud resource abstraction which constitute a SLA are different for all the three categories. Hence, these should be carefully evaluated at the time of defining SLA. Besides service models, information on deployment models should also be included. SLAs should contain the type of deployment model and deployment technologies that have been adopted by the provider.

4. Identify Critical performance Objectives: Those performance attributes that are essential as per the cloud consumer must be included in SLA. These generally include availability, response time, etc. The performance statements should be measurable and documented properly in SLA and should satisfy both the consumer as well as the provider.
5. Understand disaster recovery plan: There could be any kind of disaster such as natural disaster or any man-made event that could adversely affect the cloud consumer's business. Thus, to prevent any undue losses due to disasters a cloud provider must make appropriate disaster recovery plans. The disaster recovery plans depend to a large extent on the cloud consumer, since the infrastructure or application demands varies from one consumer to another. Therefore, the disaster recovery plan of each organization varies and business level objectives have an important role to play in disaster recovery planning. Many current cloud SLAs don't provide guarantees in case of SLA violation due to natural disaster. The cloud consumer make clear following issues regarding disaster recovery with the provider as early as possible
   - How service outage is defined?
   - What level of redundancy is adopted by cloud provider to minimize violations
   - What procedure would be followed to settle down unplanned accidents
   - What measures will be taken to prevent unplanned incidents from occurring.
   - What measures will be taken in case there is a continuous disruption in providing guaranteed services and thus causing serious impact to business.
   - What contingency plan will be adopted during a disaster
   - How the disaster recovery tests will be performed and what will be their frequency. Will the customer be informed about these tests
6. Understand the exit process: Each service level agreement must contain an exit clause. It should cite the reasons responsible for early termination and responsibilities of both the provider and consumer in case such a situation is developed.

Service level agreements help a cloud consumer to select the best cloud by matching his requirements with a cloud service level agreement. However, as the numbers of provider offering cloud service are increasing rapidly, this manual process of cloud selection becomes quite complicated and time consuming. The user may have to go through service level agreements of a number of clouds that do not even fulfil their single requirement, thus leading to wastage of their crucial time. Since, SLA of one provider varies from other provider to a great extent, reaching to a correct decision can become complicated for a user. Wrong decision can put the customer at a high risk since a customer has to risk his entire business on the provider. Thus to make the above process of selection simple for a user, this paper proposes an dynamic algorithm that would automatically select the most suitable cloud for the user and thus reduce the burden of selecting the best cloud. The cloud is selected as per user's requirements and priorities. The approach is dynamic as it will produce different results dynamically depending upon user requirements.
This paper is organized as follows. The section II discusses related work. Section III discusses the proposed work with the help of proposed algorithm. Section IV comprises of tables used in the given approach. Section V explains the results with the help of result table. The paper finally concludes in section VI.

II. RELATED WORK

Previously also an algorithm was developed with by Tejas Chauhan et al. with the motive of selecting the best cloud for a given user They developed an algorithm with the aim of selecting the best cloud provider automatically for user. It was implemented using Jena API by matching user's requirement (referred to as requirement model) with cloud's Service Level Agreement (referred to as Cloud Capability Model). The above process of matching two models was done on the basis of various service level agreement parameters Total nine parameters namely Virtual machine, Storage Capability, Memory capability, Ethernet, Availability, Processor speed, response time Server reboot, Service Credit were considered. [6]. Then a framework, SMICloud was proposed by Saurabh Kumar Garg, Steve Versteeg and Rajkumar Buyya which also selected the best cloud for the user. They used service measurement index in order to measure services of providers. AHP approach was used to rank providers. Three IaaS cloud providers namely Amazon EC2, Windows Azure, and Rackspace were taken and their performance was compared as per service measurement index using AHP ranking. [7]
Although it is very important to automate service level agreements, however in the real world, the agreements are still offered as online web pages on the provider's websites. For example Amazon EC2 SLA and GoGrid SLA are two documents that are available online on Amazon website [8] and GoGrid [16] websites respectively. [6]

III.PROPOSED WORK

The main motive of this work is to reduce the burden of cloud selection from the user. This approach would select the best cloud for the user by taking into account user requirements and priorities.
This proposed algorithm is a modification of weighted product model. According to weighted product model following equation is obtained

$$P(C_K/C_L) = \prod^{n} (C_{Kj} / C_{Lj})^{wj} \quad \text{for K, L =1, 2, 3….m} \quad [9]$$

Where,

C_k and C_L can be taken as two clouds. There are total m clouds and the ratio of all these m clouds is calculated. The ratio of two clouds is calculated by taking the product of ratios of each parameter. For example if there are 3 parameters, than $(C_{K1} / C_{L1)*} (C_{K2} / C_{L2)*} (C_{K3} / C_{L3)}$ will be calculated in order to obtain final ratio C_K/C_L. This would be done for each of the m cloud.

However, our approach will compare two clouds by taking the difference of their parameters instead of finding the ratios. Than in contrast to previous approach in which product of ratios was taken, we would add the differences obtained. The main motive of taking difference is to compare two clouds. If P (Cloud A – Cloud B) > 0, this means points of cloud A are greater than cloud B and thus cloud A is better than cloud B.

$$P(C_K - C_L) = \sum_{J=1}^{n} (C_{Kj} - C_{Lj})_{Pj} \qquad \text{for K, L =1, 2, 3…m}$$

This approach has been developed using JAVA (JDBC) as front end and My SQL as back end. It makes use of three tables:

1. Cloud provider Table -- This table contains how much service a provider promises to provide for each of the considered SLA parameter.
2. Requirement Table -- It contains the minimum amount of service that a given cloud consumer expects for each of the considered SLA parameter from the provider. Total 14 requirements are taken.
3. Priority Table – It contains the priorities of users for each of the requirements. Priorities indicate users' degree of importance for SLA parameter as per there requirements.

The proposed algorithm works as follows:

1. Match the requirement table and cloud provider table to select those clouds that meet all the user requirements. The clouds that provide service either equal to or above the values specified in requirement table will be eligible and remaining will be non eligible.
2. Calculate the points for each cloud using following formula:
   a. Points $(cloud_1) = (cloud_1 - cloud_1) = (C_{11} - C_{11}) *p1 + (C_{12} - C_{12}) *p2 + \ldots\ldots + (C_{1C} - C_{1C}) *pc \ldots\ldots + (C_{1m} - C_{1m}) *pm$
   Points $(cloud_2) = (cloud_2 - cloud_1) = (C_{21} - C_{11}) *p1 + (C_{22} - C_{12}) *p2 + \ldots\ldots + (C_{1C} - C_{2C}) * *pc \ldots\ldots + (C_{2m} - C_{1m}) *pm$
   Points $(cloud_3) = (cloud_3 - cloud_1) = (C_{31} - C_{11}) *p1 + (C_{32} - C_{12}) *p2 + \ldots\ldots + (C_{1C} - C_{3C}) *pc \ldots\ldots + (C_{3m} - C_{1m}) *pm$
   .
   .
   Points $(cloud_n) = (cloud_n - cloud_1) = (C_{n1} - C_{11}) *p1 + (C_{n2} - C_{12}) *p2 + \ldots.. + (C_{1c} - C_{nC}) *pc$ **……** $+ (C_{nm} - C_{1m}) *pm$

   The points obtained are arranged in descending order and ranks are found.
   The points can also be obtained by equations in (b) and (c). The difference between the equations in (a), (b) and (c) is that in (a) all the points are calculated with respect to $cloud_1$ (i.e. by comparing every cloud with $cloud_1$), in (b), equations points are calculated with respect to $cloud_2$ (i.e. by comparing every cloud with $cloud_2$) and in (c) points are calculated with respect to $cloud_n$ (i.e. by comparing every cloud with $cloud_n$).

   b. Points $(cloud_1) = (cloud_1 - cloud_2) = (C_{11} - C_{21}) *p1 + (C_{12} - C_{22}) *p2 + \ldots\ldots + (C_{2C} - C_{1C}) *pc \ldots\ldots + (C_{1m} - C_{2m}) *pm$ Points $(cloud_2) = (cloud_2 - cloud_2) = (C_{21} - C_{21}) *p1 + (C_{22} - C_{22}) *p2 + \ldots\ldots + (C_{2C} - C_{2C}) *pc \ldots\ldots + (C_{2m} - C_{2m}) *pm$
   Points $(cloud_3) = (cloud_3 - cloud_2) = (C_{31} - C_{21}) *p1 + (C_{32} - C_{22}) *p2 + \ldots\ldots + (C_{2C} - C_{3C}) *pc \ldots\ldots + (C_{3m} - C_{2m}) *pm$
   .
   .
   Points $(cloud_n) = (cloud_n - cloud_2) = (C_{n1} - C_{21}) *p1 + (C_{n2} - C_{22}) *p2 + \ldots.. + (C_{2c} - C_{nC}) *p3 \ldots\ldots + (C_{nm} - C_{2m}) *pm$

   The points obtained are arranged in descending order and ranks are found. The ranks are same as that obtained in (a). There are total n eligible clouds and m parameters

   c. Points $(cloud_1) = (cloud_1 - cloud_n) = (C_{11} - C_{n1}) *p1 + (C_{12} - C_{n2}) *p2 + \ldots\ldots + (C_{nC} - C_{1C}) *pc \ldots\ldots + (C_{1m} - C_{nm}) *pm$
   Points $(cloud_2) = (cloud_2 - cloud_n) = (C_{21} - C_{n1}) *p1 + (C_{22} - C_{n2}) *p2 + \ldots\ldots + (C_{nC} - C_{2C}) *pc \ldots\ldots + (C_{2m} - C_{nm}) *pm$
   Points $(cloud_2) = (cloud_3 - cloud_n) = (C_{31} - C_{n1}) *p1 + (C_{32} - C_{n2}) *p2 + \ldots\ldots + (C_{nC} - C_{3C}) *pc \ldots\ldots + (C_{3m} - C_{nm}) *pm$

.
.

Points ($cloud_n$) = ($cloud_n$ – $cloud_n$) = (C_{n1} – C_{n1}) *p1 + (C_{n2} – C_{n2}) *p2 + …..+ (C_{nc} – C_{nC}) *pc ……+ (C_{nm} – C_{nm}) *pm

The points obtained are arranged in descending order and ranks are found. The ranks are same as that obtained in (a) and (b). Step 2 would be repeated for all the requirements. Since ranks obtained by (a), (b), and (c) are same hence, any one of them can be done to calculate ranks. It is not necessary that all 3 of them have to be done.

Where,
$Cloud_1$ –1st eligible cloud
$Cloud_2$ --2nd eligible cloud
….$Cloud_n$ – nth eligible cloud. (Total there are n eligible clouds)
C_{11} – value of 1st parameter of 1st eligible cloud
C_{12} – value of 2nd parameter of 1st eligible cloud
C_{1C} – value of cost parameter of 1st eligible cloud
…. C_{1m} -- value of mth parameter of 1st eligible cloud (Total there are m parameters)
C_{21} – value of 1st parameter of 2nd eligible cloud
C_{22} – value of 2nd parameter of 2nd eligible cloud
C_{2C} – value of cost parameter of 2nd eligible cloud
.....C_{2m} -- value of mth parameter of 2nd eligible cloud
.
.
C_{n1} – value of 1st parameter of nth eligible cloud
C_{n2} – value of 2nd parameter of nth eligible cloud
C_{nC} – value of cost parameter of nth eligible cloud
…. C_{nm} -- value of mth parameter of nth eligible cloud
p_1 = priority of 1st parameter,
$p_{2=}$ priority of 2nd parameter
….$p_{m=}$ priority of mth parameter

3. Repeat above steps for all requirements.

Table I

Cloud Provider	**Rackspace**	**HP**	**GoGrid**	**Nephoscale**
Rackspace	0.0	22.55	-74.59	2036.75
HP	-22.55	0.0	-97.14	2014.20
GoGrid	74.59	97.14	0.0	2111.35
Nephoscale	-2036.75	-2014.20	-2111.35	0.0

The above table (Table I) shows the detailed points obtained by each eligible cloud for requirement1. Requirement 1 is shown in requirement table (Table III) in section IV and cloud provider table is shown in table IIs in section IV. The points are as follows:

(a) Points (Rackspace) = Rackspace− Rackspace = 0.0
Points (HP) = HP − Rackspace = −22.55
Points(GoGrid) = GoGrid−Rackspace = 74.55
Points (Nephoscale) = Nephoscale − Rackspace = −2036.75
These points (Cloud Provider -- Rackspace) are sorted in descending order and ranks are obtained.

(b) Points (Rackspace) = Rackspace –HP = 22.55
Points (HP) = HP− HP = 0.0
Points (GoGrid) = GoGrid – HP = 97.14 Points (Nephoscale) = Nephoscale −HP = −2014.20
These points (Cloud Provider – HP) are sorted in descending order and ranks are obtained (Ranks are same as obtained in (a))

(c) Points (Rackspace) = Rackspace −GoGrid = −74.9
Points (HP) = HP−GoGrid = −97.14
Points (GoGrid) = GoGrid − GoGrid =0.0
Points (Nephoscale) = Nephoscale− GoGrid = -- 2111.35
These points (Cloud Provider –GoGrid) are sorted in descending order and ranks are obtained (Ranks are same as obtained in step (a) and (b))

(d) points (Rackspace) = Rackspace− Nephoscale =2036.75
points (HP) = HP −Nephoscale =2014.20
points (GoGrid) = GoGrid − Nephoscale =2111.35
points (Nephoscale) = Nephoscale- Nephoscale =0.0
These points (Cloud Provider –Nephoscale) are arranged in descending sorted order and ranks are obtained (Ranks are same as obtained in step (a), (b) and (c))

For all the parameters such as availability, security, etc larger value is better than smaller value. For example 100% availability is better than 99% availability, 24 hours of security is better than 20 hours of security. However, for cost smaller value is better than greater value. For example, if a cloud offers services at $817.60 it is better than a cloud that offers services at $870.60. Thus, for the cost parameter whenever difference is taken between cloud A – cloud B, it is cost of cloud B – cloud A and not cost of cloud A – cost of cloud B. This would become clear by the following example:
For example, consider the case in which we have to calculate the points obtained by 2nd cloud. They can be obtained by subtracting the parameters of cloud 1from parameters of cloud 2 and then adding the differences. Now, 1st cloud offers services at $876.6 and 2nd cloud offers services at $817.6. Thus, 2nd cloud is better than 1st in cost and hence cost parameter should increase the overall points of 2nd cloud. However, like other parameters, if we subtract cost of $cloud_1$ from $cloud_2$ i.e. (817.6 – 876.6), it will result in negative answer and would decrease the points of 2nd cloud. Thus, in case of cost parameter, 2nd cloud is subtracted from 1st cloud in order to get correct answer. Similarly, when calculating points of 1st cloud, we will subtract 2nd cloud from 1st cloud for all parameters. Now, 1st cloud offers services at 876.6 and 2nd cloud offers services at 817.6. If we subtract 2nd cloud from 1st cloud i.e. (876.6 – 817.6), it will result increase the points of $cloud_1$. Thus this again justifies previous example. Therefore in case of cost parameter while calculating difference between any 2 clouds $cloud_x$ and $cloud_y$ we would calculate $cloud_y$ -- $cloud_x$ as compared to other parameters in which $cloud_x$ --- $cloud_y$ will be calculated to get correct answer.

Following service level agreement parameters are considered:
1) Security
2) Availability
3) Processor cores
4) Processor speed
5) Cost (Per hour basis/monthly basis)
6) RAM
7) Storage

IV. TABLES USED IN ALGORITHM

The above approach makes use of three tables namely cloud provider table; requirement table and priority table. These tables are stored in a My SQL database.

1) Cloud provider table contains the values of each of the considered SLA parameter that a provider aims to provide. Only those clouds that fulfil all the requirements are considered eligible and the remaining clouds are not eligible. The eligible clouds will be compared with each other and the comparisons will be multiplied with priority in order to find the best cloud. Data has been collected from websites of different providers. 10 providers are taken namely Google compute engine [10][11], Rackspace[12][13], HP[14][15], GoGrid[16][17], Opsource[18][19], Nephoscale[20][21], Bitrefinery[22][23], Windows azure[24][25], saavisdirect[26][27], Joyent[28][29]. Since no information about security was given, hence it has been assumed

TABLE II
CLOUD PROVIDER TABLE

Cloud Provider	**Security**	**Availability**	**Processor speed(per core)*(approx)**	**Processor Cores**	**Cost (per hour basis)**	**Cost (monthly basis)**	**RAM**	**Storage**
Google compute engine	22 hours	99.95%	Not Mentioned	8	$1.06	Not Mentioned	30 GB	3540GB
Rackspace	23 hours	100%	2.3 GHz	8	$1.20	$876.6	30 GB	1228GB
Hp	22 hours	99.95%	2.7 GHz	8	$1.12	$817.6	32 GB	960 GB
GoGrid	24 hours	100%	2.9 GHz	24	$1.92	$870	24 GB	1228 GB
OpSource	22 hours	100%	2.1 GHz	8	$2.17	$1584.10	64 GB	2500GB
Nephoscale	22 hours	99.95%	2.4 GHz	8	Not Mentioned	$1499.00	144 GB	1000 GB

Bitrefinery	23 hours	100%	2.1 GHz	4	Not Mentioned	$246.20	8 GB	150 GB
Windows azure	22 hours	99.95%	1.6 GHz	8	$1.80	$1399	56 GB	2040 GB
Savvisdirect	22 hours	99.9%	2.67GHz	8	Not Mentioned	$329.87	8 GB	500 GB
Joyent	22 hours	100%	Not Mentioned	16	$2.80	$2044	80 GB	2048GB

2) User Requirement Table -- The 2nd table is user requirement table. It consists of how much a user expects for different parameters from providers. The value of each parameter (except cost) is the minimum value that a given user expects from providers. However, in case of cost, the value shown in table is maximum budget of the user. For example 20 hours security means minimum security user expects is 20 hours and every cloud that provides security either equal or above that value will be selected. However, $2000 means maximum cost that a user is willing to pay is $2000 and every cloud that offers service below the given cost will be selected. This table is matched with the cloud provider table in order to find the eligible clouds. If a cloud fails to provide service above that mentioned in the requirement table even for a single parameter, it is considered as non eligible. The eligible clouds are ranked using the above algorithm.

TABLE III
USER REQUIREMENT TABLE

Requirements	**Security**	**Availability**	**Processor cores**	**Processor speed(per core)*(approx)**	**Cost**	**RAM**	**Storage**
Requirement1	20 hours	97.5%	8	2.2 GHz	$2500	20 GB	600 GB
Requirement 2	19 hours	98.5%	8	2.3 GHz	$2700	25GB	500 GB
Requirement 3	Not Required	94.5%	4	2 GHz	$2500	Not Required	100 GB
Requirement 4	Not Required	90.5%	4	1.8GHz	$1700	20 GB	400 GB
Requirement 5	20 hours	100%	4	2.1GHz	$2(Per Hour)	8GB	800GB
Requirement 6	22 hours	90%	8	1.5 GHz	$2200 (Per month	16 GB	400 GB
Requirement 7	Not Required	94.0%	Not Required	Not Required	$3(Per hour	Not Required	Not Required
Requirement 8	20 hours	90%	4	2.1 GHz	$2100 (Per month)	8 GB	100 GB
Requirement 9	17 hours	100%	8	1.7GHz	$2000 (Per month)	8 GB	400 GB
Requirement 10	20 hours	Not Required	8	Not Required	$1500 (Per month)	16 GB	900 GB
Requirement 11	22 hours	Not Required	Not Required	Not Required	$2900(Per month)	32 GB	Not Required
Requirement 12	Not Required	Not Required	Not Required	Not Required	$2.8(Per hour)	Not Required	100 GB
Requirement 13	Not Required	Not Required	4	Not Required	$3 (Per month)	Not Required	1000 GB
Requirement 14	Not Required	100%	8	Not Required	$2500 (Per month)	16 GB	100GB

3) Priority Table -- Priority table depicts importance of different Service level agreement parameters for users. If two or more parameters have same importance, they will be assigned same priority. These priorities can be thought of as weights that a user assigns to different parameters. The main motive of prioritizing different parameters is to make sure that the cloud assigned to a user provides good service for those parameters that are important for a user. The priorities are given in descending order i.e. the most important parameter is given highest in number priority. For example in requirement 1, most important parameter is availability, thus it is assigned priority 7th. This is so because these priorities will be multiplied with difference to find points and the cloud getting highest points is best. Hence,

these priorities are arranged in descending order, with the highest priority given to that parameter that is most important.

TABLE IV
PRIORITY TABLE

Requirements	**Security**	**Availability**	**Processor cores**	**Processor speed(per core)**	**RAM**	**Storage**	**Cost**
Requirement 1	3rd	7th	5th	2nd	6th	1st	4th
Requirement 2	3rd	2nd	4th	5th	1st	7th	4th
Requirement 3	0	3rd	1st	3rd	0	2nd	4th
Requirement 4	0	3rd	3rd	4th	1st	2nd	4th
Requirement 5	1st	6th	6th	2nd	4th	3rd	5th
Requirement 6	5th	3rd	4th	2nd	1st	0	1st
Requirement 7	0	1st	0	0	0	0	2nd
Requirement 8	5th	5th	1sts	6th	3rd	4th	2nd
Requirement 9	3rd	6th	5th	4th	2nd	1st	7th
Requirement 10	4th	0	4th	0	1st	2nd	3rd
Requirement 11	2nd	0	0	0	3rd	0	1st
Requirement 12	0	0	0	0	0	2nd	1st
Requirement 13	0	0	1st	0	0	3rd	2nd
Requirement 14	0	3rd	2nd	0	4th	1st	1st

V. RESULTS

The following table shows the results obtained by applying above algorithm

TABLE V
RESULT TABLE

REQUIREMENTS	**Google Compute Engine**	**Rackspace**	**HP**	**GoGrid**	**Opsource**	**Nephoscale**	**Bit Refinery**	**Windows Azure**	**Savvisdirect**	**Joyent**
Requirement 1	Not Eligible	2nd	3rd	1st	Not Eligible	4h	Not Eligible	Not Eligible	Not Eligible	Not Eligible
Requirement 2	Not Eligible	1st	2nd	Not Eligible	Not Eligible	3rd	Not Eligible	Not Eligible	Not Eligible	Not Eligible
Requirement 3	Not Eligible	4th	6th	3rd	5th	7th	2nd	Not Eligible	1st	Not Eligible
Requirement 4	Not Eligible	2nd	4th	1st	3rd	5th	Not Eligible	Not Eligible	Not Eligible	Not Eligible
Requirement 5	Not Eligible	2nd	Not Eligible	1st	Not Eligible	Not Eligible	Not Eligible	Not Eligible	Not Eligible	Not Eligible

Requirement 6	Not Eligible	3rd	2nd	1st	6th	5th	Not Eligible	4th	Not Eligible	Not Eligible
Requirement 7	1st	4th	3rd	6th	7th	Not Eligible	Not Eligible	5th	Not Eligible	Not Eligible
Requirement 8	Not Eligible	3rd	4th	2nd	1st	5th	7th	Not Eligible	6th	Not Eligible
Requirement 9	Not Eligible	2nd	Not Eligible	1st	3rd	Not Eligible	Not Eligible	Not Eligible	Not Eligible	Not Eligible
Requirement 10	Not Eligible	3rd	4th	1st	Not Eligible	5th	Not Eligible	2nd	Not Eligible	Not Eligible
Requirement 11	Not Eligible	Not Eligible	1st	Not Eligible	4th	2nd	Not Eligible	3rd	Not Eligible	5th
Requirement 12	1st	4th	6th	5th	2nd	Not Eligible	Not Eligible	3rd	Not Eligible	Not Eligible
Requirement 13	1st	6th	Not Eligible	5th	2nd	Not Eligible	Not Eligible	4th	Not Eligible	3rd
Requirement 14	Not Eligible	3rd	Not Eligible	2nd	1st	Not Eligible	Not Eligible	Not Eligible	Not Eligible	4th

The above table shows the ranks obtained on applying the above algorithm. Not eligible means that given cloud does not meet all the requirements.

In Requirement 1 only four clouds namely Rackspace, HP, GoGrid and Nephoscale are eligible and remaining are ineligible. The reasons for the same are explained as follows:

Google compute engine does not provide service on monthly basis as required by 1st requirement and has no mention of processor speed (2.1 GHz speed is required by Requirement 1). Opsource does not provide 2.2 GHz processor speed and hence is not eligible. Bit refinery does not provide required RAM (20 GB), storage (600 GB) and processor speed (2.2 GHz). Windows Azure engine does not to provide required processor speed (2.2 GHz). Saavisdirect does not provide required RAM (20 GB), storage (600 GB). Joyent also does not provide 2.2 GHz processor speed. All the remaining clouds are eligible and are ranked according to the algorithm discussed above out of which GoGrid achieves the first rank. Similarly results can be obtained for other requirements also.

VI. CONCLUSIONS

The approach developed in this paper ranks all the clouds providers and finds the best cloud provider for the user. The same work when done manually by a consumer becomes quite tedious and difficult. Hence, this approach makes the work of provider selection quite simple for a user by automatically selecting the best cloud for a user as per his requirements and priorities. In spite of the fact that this task of provider selection is quite complicated when performed by a consumer manually, there is hardly any automatic selection of cloud providers in real world and the service level agreements are still offered as manual documents on websites of various providers. Hence, this approach provides an answer to one of the drawbacks faced by cloud computing. It also helps the provider to identify its weakness by comparing its services with the services offered by other clouds in market. We hope that this effort of ours will help those working in the area of cloud computing with their future works.

REFRENCES

[1] Goudarzi, Hadi, Mohammad Ghasemazar, and Massoud Pedram. "SLA-based optimization of power and migration cost in cloud computing." in *Proc* Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on. IEEE, 2012.

[2] The Wikipedia website(online) http://en.wikipedia.org/wiki/Cloud_computing

[3] Lee Badger, Tim Grance , Robert Patt-Corner and Jeff Voas, Cloud Computing Synopsis and Recommendations, NIST Special Publication 800 (2012):146.

[4] "Cloud"; SLAs for cloud service ETSI TR 103 125 V1.1.1 (2012-11).Technical report
[5] Practical Guide to Cloud Service Level Agreements ,cloud standard customer council, version 1.0April 10, 2012
[6] Tejas Chauhan, Sanjay Chaudhary, Vikas Kumar, and Minal Bhise , "Service Level Agreement parameter matching in Cloud Computing, Information and communication technologies (WICT)", in *proc*. 2011 World congress on IEEE, 2011.
[7] Saurabh Kumar Garg, Steve Versteeg and Rajkumar Buyya, "SMICloud: a framework for comparing and ranking cloud services." In proc. Utility and Cloud Computing (UCC) 2011 Fourth IEEE International Conference on. IEEE, 2011.
[8] The Amazon website (online). Available: http://aws.amazon.com/ec2-sla/
[9] The Wikipedia website (online). Available: http://en.wikipedia.org/wiki/Weighted_product_model
[10] The Google website (online). Available: https://developers.google.com/compute/docs/sla
[11] The Google website (online). Available: https://cloud.google.com/pricing/compute-engine
[12] The Rackspace website (online). Available: http://www.rackspace.com/information/legal/cloud/sla
[13] The Rackspace website (online). Available: http://www.rackspace.com/cloud/servers/pricing/
[14] The HP website (online). Available: https://www.hpcloud.com/SLA
[15] The HP website (online). Available: https://www.hpcloud.com/pricing
[16] The GoGrid website (online). Available: http://www.gogrid.com/legal/service-level-agreement-sla
[17] The GoGrid website (online). Available: http://www.gogrid.com/products/pricing
[18] The Opsource website (online). Available: http://www.opsource.net/Services/Cloud-Hosting/Service-Level-Agreement
[19] The Opsource website (online). Available: http://www.opsource.net/Services/Cloud-Hosting/Pricing
[20] The Nephoscale website (online). Available: http://www.nephoscale.com/service-level-agreement
[21] The Nephoscale website (online). Available: http://www.nephoscale.com/dedicated-servers
[22] The Bit refinery website (online). Available: http://bitrefinery.com/tos
[23] The Bit refinery website (online). Available: http://bitrefinery.com/cloud-hosting/pricing
[24] The Microsoft website (online). Available: http://www.microsoft.com/en-us/download/details.aspx?id=38427
[25] The Microsoft website (online). Available: http://www.windowsazure.com/en-us/pricing/calculator/
[26] The savvisdirect website (online). Available: http://www.savvisdirect.com/slas
[27] The savvisdirect website (online). Available: http://www.savvisdirect.com/cloud-servers/pricing
[28] The Joyent website (online). Available:http://joyent.com/company/policies/cloud-hosting-service-level-agreement
[29] The Joyent website (online). Available: http://joyent.com/products/joyent-cloud/pricing

