International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 32371-32385

http://www.ripublication.com

# Trust Based Ranking Service Models in Cloud Computing Environment

**M.Saravanan[1] And M.Aramudhan[2]**

*Assistant Professor, Department of CSE,*
*Periyar Maniammai University, Thanjavur, India*
*e-mail : saran84gct@gmail.com*
*Associate Professor, Department of IT,*
*Perunthalaivar Kamarajar Institute of Engineering and Technology, Karaikal, India*
*e-mail: aramudhan1973@yahoo.com*

## Abstract

Cloud Computing is a growing and excellent technology where user can access services as required as much he needs from service providers through internet from anywhere at any time. As rapidly increasing the interest among users to utilize applications and services in the internet world, they are approaching cloud service providers, so service providers also rapidly increasing in wide range, this leads ambiguity and distrust among the customers. One of the challenges faced by potential cloud customers is "how to distinguish the best fit cloud services provider to meet their anticipation". Our proposed frame work will address few key concepts to select the appropriate service provider for their requirements. We proposed grade table for categorize the resources, grade values and grade distribution algorithm to distinguish components, grade total computation for rank the service providers and priority based decision tree applied to separate best service provider among similar highest rank list. Our frame work also provides trustworthy environment to the user by evaluating output with help of trust evaluation unit. We used grades are Gold, Silver and Bronze to categorize the resources, grade values 1.0, 0.5, and 0.25, for represents key components.

**Keywords**: Broker Architecture, Grade Distribution Table, Grade distribution algorithm, Priority Based Decision Tree.

**Introduction**

Cloud computing is an interconnected computing resources to provide on demand access basis to the user, (e.g., infrastructure, platform, and software) Cloud computing identifies "five essential characteristics":[1] On *demand self-service.* A consumer made self provision computing capabilities, such as resource provider and network storage, as needed automatically without requiring human interaction with each service provider.[1]

*Wide network access*: Resources are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., Tablets, laptops, mobile phones, and workstations).

*Resource pooling:* The provider's computing resources are pooled to serve multiple consumers using a multi-pace model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.[1]

*Rapid scalability:* Services can be elastically provisioned and distributed, in some cases automatically, to scale rapidly outward and inward proportionate with demand. To the consumer, the resources available for provisioning often appear unlimited and can be appropriated in any quantity at any time. [1]

*Measured service:* Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, speed, bandwidth, and dynamic user accounts). Resource usage can be examined, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.[1]Public clouds are managed by public cloud service providers, which include the public cloud environment's servers, storage, and networking data center operations. Users of public cloud services can generally select from three basic categories:

*User self-provisioning*: Customers buy cloud services directly from the provider, typically through internet. The customer pays on a per-transaction basis.[2]

*Advance provisioning*: Customers contract in advance a predetermined amount of resources, which are prepared in advance of service. The customer pays a usage fee or a monthly fee.[2]

*Dynamic provisioning*: The provider assigns resources when the customer wants them, then disconnect them when they are no longer needed. The customer is charged on a pay-per-use basis[2].Best resource offering (Calheiros, Vecchiola, Karunamoorthy, & Buyya, 2011), Automatic Selection and Ranking of Cloud Providers using Service Level Agreements(PreetiGulia, SumedhaSood, 2013), Ranking model for SLA resource provisioning management(C.S.Rajarajeswari, M.Aramudhan, 2014), papers discussed about resource provisioning to user, based on their requirements.

**1. Related Work**

**i. Functions of Broker manger**

Johan Tordsson a, ∗, Rubén S. Monterob, Rafael Moreno-Vozmedianob, Ignacio M. Llorente b[3] (29 July 2011) Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers In this discussion, the use of optimized

cloud brokering mechanisms are essential to transform the heterogeneous cloud market into a commodity like service. These cloud brokers have a two folded role. First, they provide the scheduling mechanisms required to optimize distribution of VMs amongst more than one clouds. Second, they offer a uniform management interface with operations, e.g., to deploy, monitor, and terminate VMs, with autonomy of the particular cloud provider technology.

**ii. Ranking Techniques**

PreetiGulia, SumedhaSood[4](2013) proposed technique was dynamic ranking and selection of cloud providers using Service Level Agreements. The selection and ranking of clouds is done by matching user requirements with Service Level Agreements offered by different clouds according to user assigned weights.

Saurabh Kumar Garg a, ∗, Steve Versteeg b, Rajkumar Buyyaa A framework for ranking of cloud computing services. (2013) Proposed framework and a mechanism that measures the quality and prioritize Cloud services. Such a framework can make a significant impact and will create healthy competition among Cloud providers to satisfy their Service Level Agreement (SLA) and improve their QoS. Different Techniques used Multiple Attribute Utility Theory (MAUT), Outranking Method, and Analytic Hierarchy Process. In this work user assigns weight to each QOS service attributes.

**iii. Broker Manger and Ranking Techniques.**

C. S. Rajarajeswari, M. Aramudhan, [5] (2014) Ranking Model for SLA Resource Provisioning Management. Presented work provides a new federated cloud mechanism, in which Cloud Broker Manager (CBM) takes up the responsibility of resource provisioning and ranking. In this technique they used pCBM and sCBM to handle sla and non sla user. Dynamic Loose Priority Scheduling used to mange queue, Markov process used to map the available services from service provider and finally they used Poincare plot mechanism for ranking the service providers.

## 2. Proposed Work

Our proposed frame work addresses some key concepts to select the appropriate service provider based on the user requirements. We proposed grade table for categorize the resources, grade values to distinguish components, grade total computation for rank the service providers and priority based decision tree applied to separate best service provider among similar highest rank list. Our frame work also provides trustworthy environment to the user by evaluating output with help of trust evaluation unit. We used grades are Gold, Silver and Bronze to categorize the resources, grade values 1.0, 0.5 and 0.25, used to assign into key components of user requests. In our discussions KPI's(key performance indicators) used to evaluate the cloud service providers. Service models in cloud categorized as follows Infrastructure as a service (like Amazon/ Rackspace ), SaaS, platform as service PaaS (like SalesForce.Com, Windows Azure).

**Table-1**

| Service-level Types | KPIs | Definition | Measurement In Unit |
|---|---|---|---|
| **Availability** | Service window | Measurements of KPI with time duration. | Time range |
| | Service/System availability | Availability defined the time service is accessible by the user. Percentage of time that service or system is available. | % |
| | MTBF | Meantime between failure | Time units |
| | MTTR | Meantime to repair | Time units |
| **Cost** | User self provisioning | Pay per-transaction basis | $ |
| | Advanced Provisioning | The customer pays a flat fee or a monthly fee | $ |
| **Performance** | Response time | Response time for composite or atomic service | Seconds |
| | Elapsed time | Completion time for a batch or background task | Time units |
| | Throughput | Number of transactions or requests processed per specified unit of time | Transaction or request count |
| **Capacity** | Bandwidth | Gross capacity of the connection. Amount of data which cloud be transmitted within a time unit | Bps |
| | Processor speed | Clock speed of a processor | MHz |
| | Storage capacity | Capacity of a temporary or persistent storage medium, such as RAM, SAN, disk, or tape | GB |
| | Storage Types | RAID levels 0, 1, 2, 5, 6 | Cost |
| **Reliability** | Service/System reliability | Probability that service or system is working flawlessly over time | % |
| **Scalability** | Service/System scalability | Degree to which the service or system can support a defined growth scenario | Yes/No, or description of scalability upper limit |
| **Security** | Type of Security | Low, Medium, High level | Rate |
| | Audit | scope and frequency of security audits | Count |
| | Back up | Infrastructures service, Storage as service for back up the content. | MB |

| | | | |
|---|---|---|---|
| **Licensed** | Software | Updated version, Timeliness of antivirus software and the isolation and logging. | Updated |
| **Speed** (Average speed) | Read speed | Average reading speed usually refers to an individual hard drive. This value indicates how fast data can be read from the hardware. In RAID systems | Seconds |
| | Write speed | Just like the reading speed it refers to the write speed to the hard drive. This value thus indicates how quickly data can be written from a source to the hardware. | |
| Provisioning Type | Thin provisioning | Client gets the storage not permanently assigned by service provider but it is dynamically allocated at runtime | Range |
| | Thick provisioned | Storage is allocated to the customer immediately. | |

## 3. Cloud Federated Broker Architecture

Enhanced federated cloud model categorizing user as SLAuser and nonSLA user, Register with particular service provider or specifying particular service provider come under the category of SLA user and non register with any of service provider named as nonSLA user. User requirement classified as functional SLA parameters includes memory related information like memory size, CPU cores, CPU size etc. and non-functional SLA parameters consists of response time, cost, execution time, security etc (Jrad, Tao, & Streit, 2012; Aljawarneh, 2011).and grade table used for rank the Cloud service providers and Trust evaluation unit used for verify the outcome of service provider.Cloud Federated Broker architecture functionality involves three phases. They are Discovery, Allocation & Monitoring and Marketing. (Calheiros, NadjaranToosi, Vecchiola, & Buyya, 2012; Grozev & Buyya, 2012).[9]

Brokers obtain the request from users and perform services based on functional and non-functional parameters, such as Key Performance Indicators [6]. In the proposed model, Federated Cloud Broker Manager (FCBM) communicates with service provider broker. FCBM, There are two types of CBM namely slaCBM and nonslaCBM.

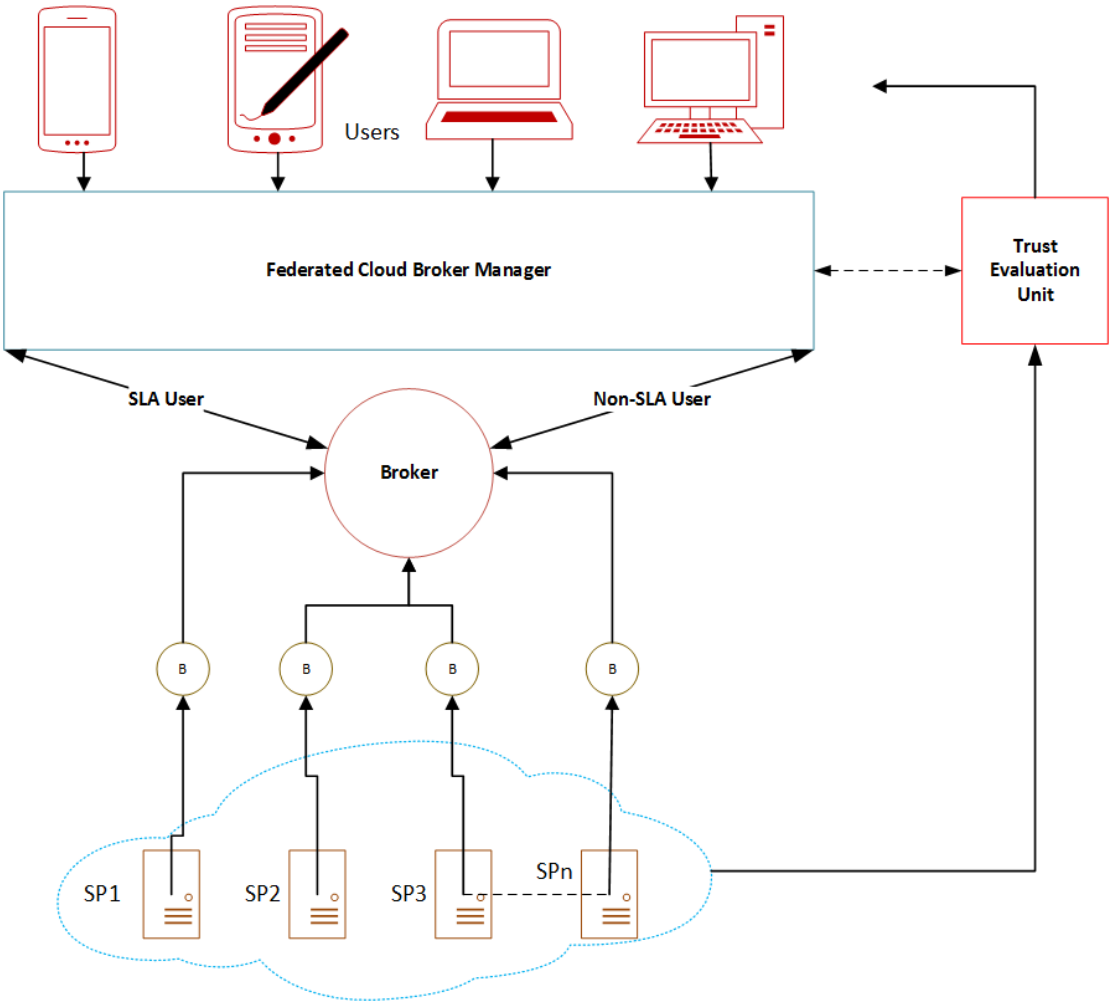


**Figure 1**

The roles of slaCBM in the proposed cloud architecture are (i) Obtaining request from the SLA user.(ii) Match the requirements of user with registered service provider (iii) Executing the task with that service provider (iv) Invoking the Trust Evaluation Unit.

**Algorithm1**

nonslaCBM has proposed to provide services to nonsla user or not specifying particular service provider. Broker will be given better path to choose suitable service provider to user.

Broker Management Algorithm
Step 1:

Federated CBM receives a request from user.
Step 2:

FCBM verify that user category: SLA or nonSLA member.
If user is **SLA member**
**Then**
Submitted to **slaCBM** and process the task
**Else**
Submitted **to nonslaCBM** and process the task.
Step 4:
Invoke Trust Evaluation Unit.
End procedure

The function of nonslaCBM is as follows (i) Get user requirements, ii) List available service providers iii) Assign grades to each service providers based on the user requirements, iv) Rank the service providers based on the grades, v) Assign best service provider to the user and Execute the task, vi) Invoke Trust evaluation unit.

Speed, security type etc...) when greater than user requirement specified in the request. Such a way 'Silver' value offered to KPI when the requirements approximately equal. 'Bronze' value offered to the resource parameter when less amount of resources offered by the service provider.

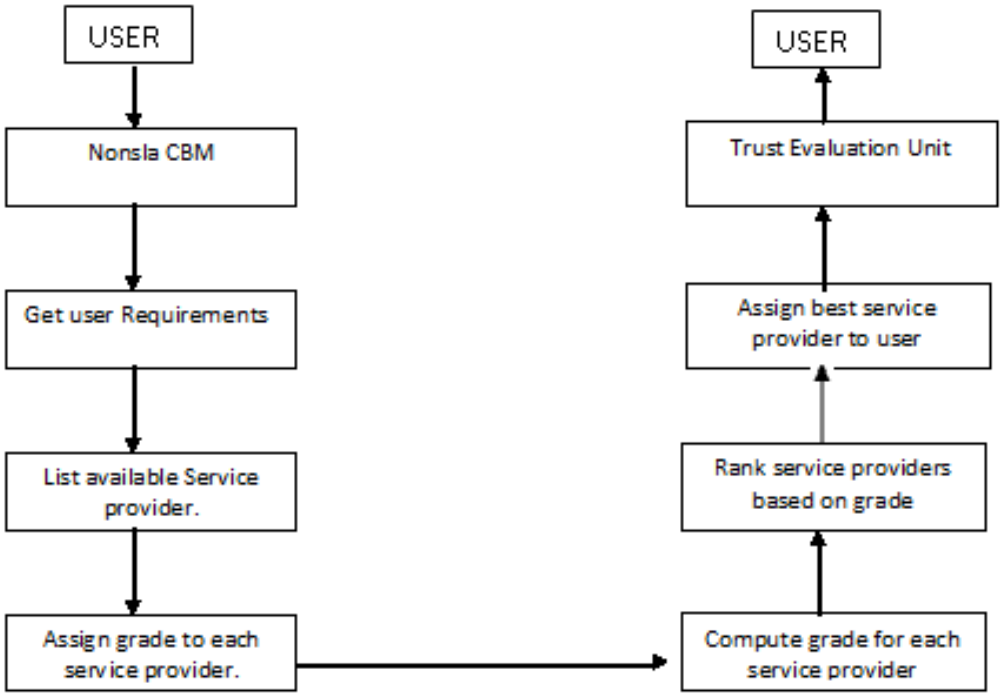


**Figure2**

4. **Ranking Methodology**

**4.1 Grade Table**

We proposed grade table and grades are used as follows Gold, Silver, and Bronze. Each grade assigned with numerical values (1, 0.5, and 0.25). These values are distributed to the service providers of resources, requested by the customer (KPI) requirements. 'Gold' value offered to the key performance indicator (like availability, memory type, and processor

**Table 2**

| Serial. No | Grade | Numerical Value(G) |
|---|---|---|
| 1 | Gold | 1.00 |
| 2 | Silver | 0.50 |
| 3 | Bronze | 0.25 |

### 4.2 Grade Distribution Fuzzy algorithm-2

Algorithm 2: nonslaCBM
Step 1:
Get the user requirement parameter: $R_{i\ (i=1, 2, 3, 4, 5….n)}$
Step 2:
List out the available Service Provider based on $R_{i\ (i=1, 2, 3, 4, 5….n)}$
Step 3:
Assign Grade Value based on user Requirement Parameter $R_{i\ (i=1, 2, 3, 4, 5….n)}$
If (Resource Available > Requirement)
Assign 'Gold' Grade value=1;
If (Resource Available < Requirement)
Assign 'Bronze' Grade value=0.25;
If (Resource Available == (approximately)Requirement)
Assign 'Silver' Grade value=0.5;
Elseif
(Resource Available== 'Nil' )
Assign Grade value=0;
Step 4:
Compute Grade total to for each service provider using formula 2.
Step 5:
Rank the service providers based on the grade total.
Step 6:
Assign the best service provider to the user.
Step 7:
Invoke Trust Evaluation Method TEM();
End procedure.

### 4.3 Grade Computation Method

Using Grade Distribution Algorithm, allocate grades value to each component required by the user, after assign grade values need to compute total grade value for each service provider using formula Service provider Grade total $[SG]_{total}$. **N** represents number of components required for user request.

$$[SG]_{total} = \sum_{k=1}^{N} G_k$$

Available service providers listed based on the requirement of user request. 'n' Represents number of service providers(n) available in the list.

$$S= \{S1, S2, S3, S4, \ldots\ldots\ldots\ldots\ldots..S_n\}$$

$$[SG_{total}]_n = \sum_{k=1}^{N} G_k$$

### 4.4 Service Providers Ranking Method

User request received and extracted the required components by the broker manager, Figure 2 shows Grade distributed to the each available components of the service providers based on the user requirements. Now grade table maintaining the service providers grade list, from the grade table, grade total $[SG]_{total}$ and grade mean $[SG_{total}]_{mean}$ to be computed and stored in table. Service providers are ranked based on their grade total mean $[SG_{total}]_{mean}$ using Quick sort data structures algorithm. After sorting the $[SG]_{mean}$ values, top service provider identified and assigned to user, if more than one service providers on similar category of top rank, then the selection process redirected to priority based decision tree(PBDT) technique.

Quicksort (sometimes called partition-exchange sort) is an competent and best sorting algorithm, serving as a systematic method for placing the elements of an array in order. Mathematical of quicksort shows that, on average, the algorithm takes O($n$ log $n$) comparisons to sort $n$ items. In the worst case, it makes O($n^2$) comparisons, though this behavior is rare, So our frame work we used quick sort technique to arrange grade mean values. After find the service grade total for each service providers, Individual Service Provider grade mean $[SG]_{mean}$ need to find for rank the service providers.

$$[SG]_{mean} = [\frac{\sum_{k=1}^{N} G_k}{N}]$$

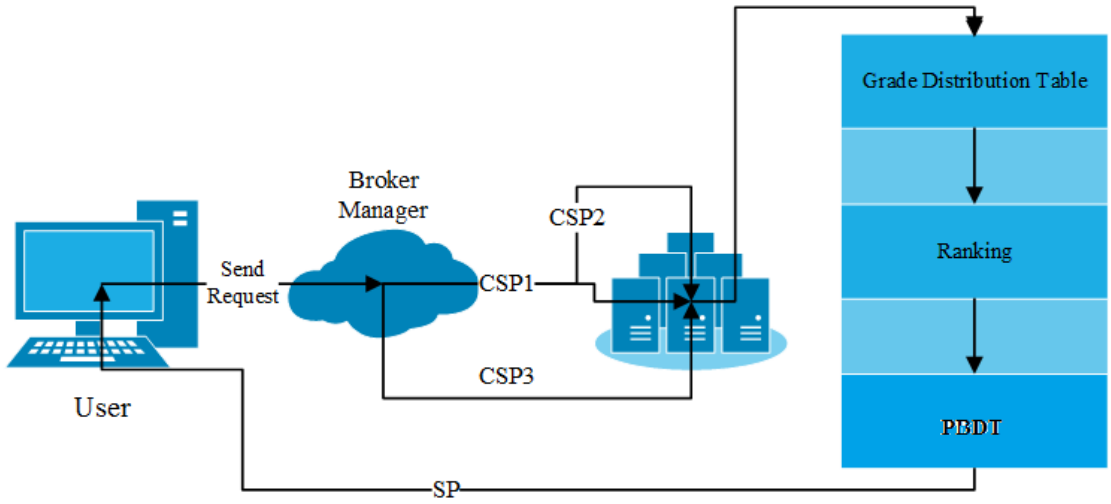


**Figure 2**

**5. Priority Based Decision Tree**

Decision trees are powerful and popular tools for classification and prediction. Decision trees represent *rules*, which can be understood by humans and used in knowledge system such as database. We framed rules for select best service provider from the similar highest rank grade total mean posses by the service provider list. Priority components are the input submitted to the decision tree. Decision tree introduced two cases in our discussion. Primary level priorities (α), Secondary level priorities (β), Former one is the highest priority where all parameters of user request being computed maximum grade provided to the service provider. Later one is less than or equal to silver grade value.

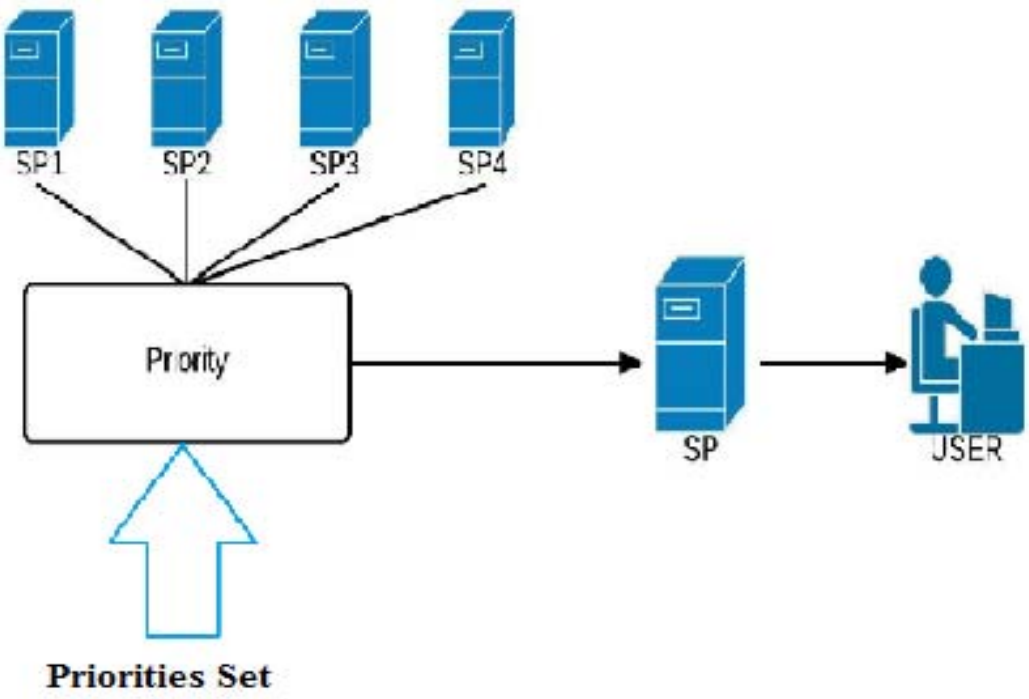


**Figure 3**

**Case 1:**

Primary level priority set included from grade table and grade values assigned to individual service providers. Grade values (G) assigned to Priority (P), user allowed to choose (N) number of priorities, if all priorities are computed, then maximum value of (α) is one. When α=1 service provider assumed to be best service provider. Below equation 'w' represents maximum number of service providers to be reach from first service provider.

$$[\alpha]_w = \prod_{i=1}^{N} P_i, \alpha = 1, P \neq 0$$

In β equation 'y' represents maximum numbers of service provider from the secondary level priority set. (N) Represents user level priorities.

$$[\beta]_y = \prod_{i=1}^{N} P_i, \beta \leq 0.5, P \neq 0$$

Priority grade values (P) should not zero for any component of user submitted component.

**Case 2:**
Secondary level priority set derived from primary level priority set, any one of the component grade value is less than or equal to 0.5, then the service provider assigned to secondary level priority decision (β), Decision will be executed only $\alpha \neq 1$.Secondory level priority set help to user to select next top service provider.

## 6. Simulation Results and Summary

User requirements extracted from request and processed based on the grades defined from the table and algorithm. In our discussion we considered four service providers Available Resource (AR), Grade (Gr) values allocated by the fuzzy based grade distribution algorithm. Quick sort algorithm used to evaluate best service provider.

**Table 3**

| User Requirements (types) | Attributes Requirements | SP1 | | SP2 | | SP3 | | SP4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | AR | Gr | AR | Gr | AR | Gr | AR | Gr |
| Processor speed | 2.4 GHz | 1.8 GHz | 0.25 | 2.4 GHz | 0.5 | 2.45 GHz | 1.0 | 2.1 GHz | 0.25 |
| Availability | 99.9% | 90.0% | 0.25 | 99.0% | 0.5 | 100% | 1.0 | 95% | 0.25 |
| Memory (RAM) | 4GB | 4GB | 1.0 | 4GB | 1.0 | 4GB | 1.0 | 4GB | 1.0 |
| Service response time | 60-100s | 60-120s | 0.5 | 60-100s | 1.0 | 60-120s | 0.5 | 40-120s | 0.25 |
| Security | High | Medium | 0.5 | Medium | 0.5 | High | 1.0 | Medium | 0.5 |
| Grade Total | | | 2.5 | | 3.5 | | 4.5 | | 2.25 |
| Grade Mean | | | 0.5 | | 0.7 | | 0.9 | | 0.45 |

From above table maximum grade mean value $[SG]_{mean}$ found for Service Provider(S3), So service provider S3 considered to be a best service provider among all service provider. Graph 1 shows same in graphical representation.

Graph 2 shows ranked service provider list top service provider to lowest service provider listed over in this graph. If more than one top rank service providers available, then process redirected to priority based decision tree.

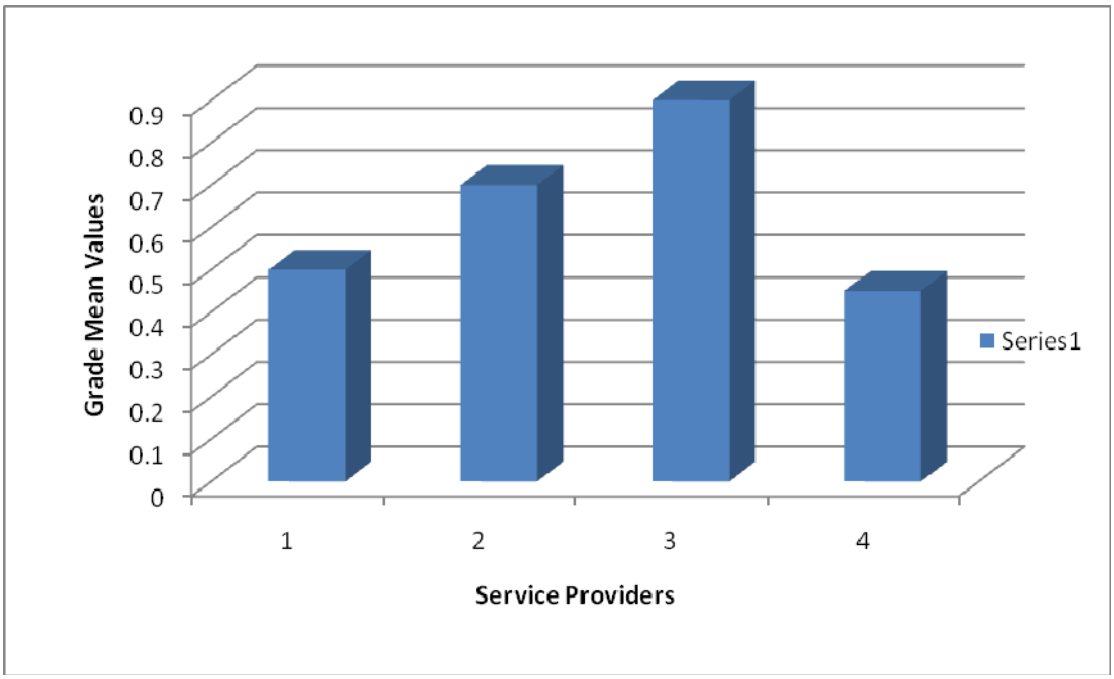


**Figure 4-(Before Rank 1)**

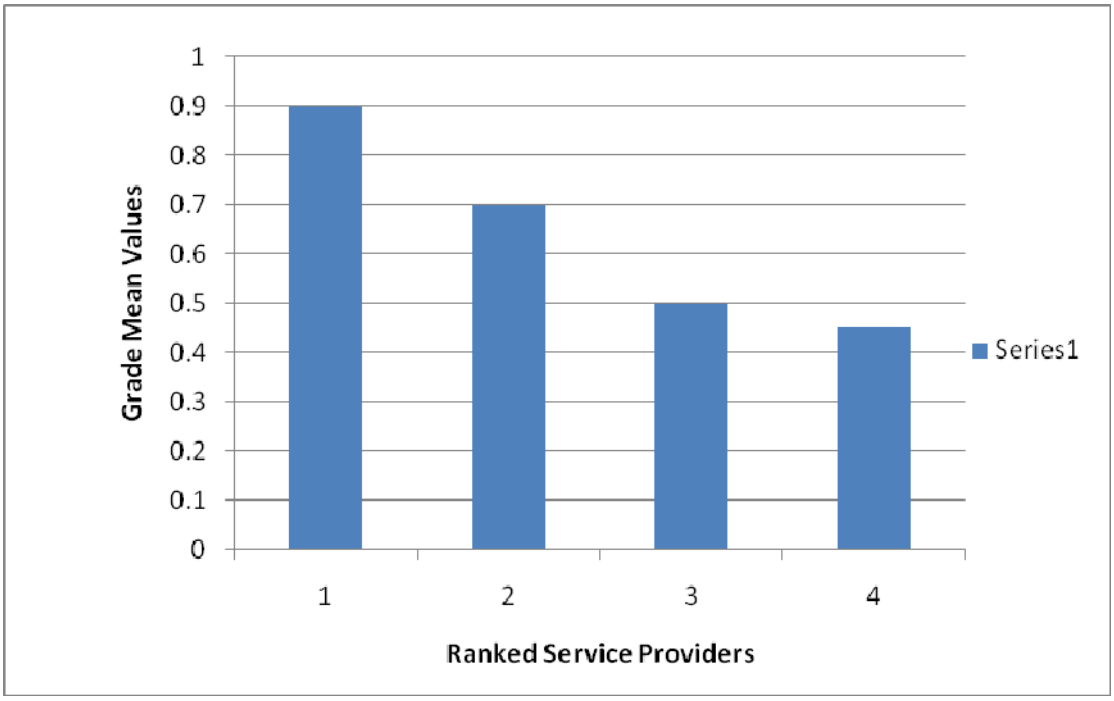


**Figure 4.1**

**7. Trust Evaluation Method**

Trust evaluation unit provides trust environment to user by review output driven by service provider. Service level agreement parameters used in SLA table used to compare with output response. We fixed deviation threshold, tolerable threshold value is 1-10%,

If negative threshold value obtained then user assumes SLA violation from service provider.

$$Deviation(Threshold) = \frac{SLA(input) - Output}{SLA(input)}$$

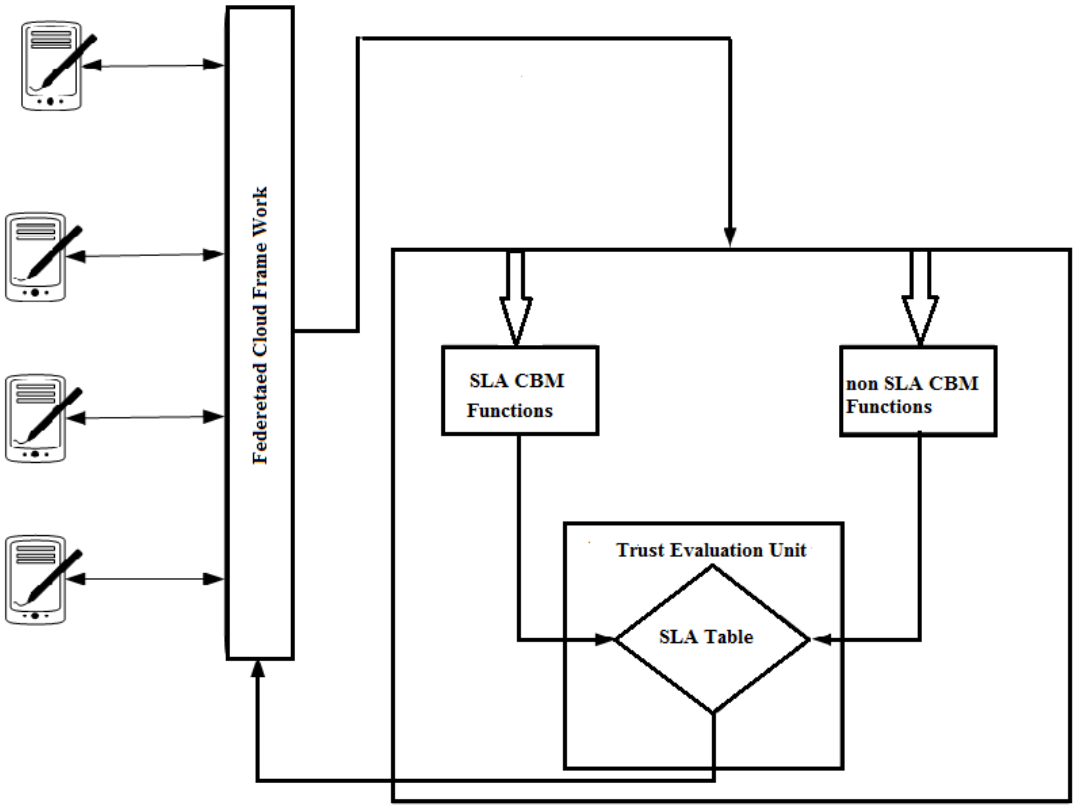


**Figure 5**

**Conclusion and Future Work**

Cloud Computing has become an emerging and excellent technology where user can remotely access any kind of services as much as he needs from service providers through internet. As rapidly increasing the interest among users to utilize application and services the in the cloud world, Service providers also increasing in wide range, this leads ambiguity and distrust among the users to select the best service provider

for their requirement. Therefore many authors proposed frame work to select best service providers based on the user requirement, relative ranking etc...Our proposed frame work help to user to select the appropriate service provider with aid of grade computation and priority based decision tree. Each service provider assigned grades and based on the grades service providers will be ranked, from the rank top service provider assigned to user based on his requirements. Also our frame work provides trustworthy environment to the user by evaluating output with help of trust evaluation unit. In future we try extending the grades to dynamically changing requirements of user and dynamically compute trust values for customer requirements.

## References

1. National Institute of Standards and Technology
2. http://en.wikipedia.org/wiki/Cloud_management
3. Jens Happe, Wolfgang Theilmann, Andrew Edmonds, and Keven T. Kearney "A Reference Architecture"for Multi-Level SLA Management Springer Science+Business Media, LLC 2011
4. Preeti Gulia, Sumedha Sood (2013) 'Dynamic Ranking and Selection of Cloud Providers Using Service Level Agreements' International Journal of Computer Applications (0975 – 8887) Volume 72– No.11, May 2013
5. C. S. Rajarajeswari, M. Aramudhan, (September 2014) Ranking Model for SLA Resource Provisioning Management International Journal of Cloud Applications and Computing, 4(3), 68-80, July-September 2014
6. Stefan Frey, Claudia L¨uthje, Christoph Reich 'Key Performance Indicators for Cloud Computing SLAs': The Fifth International Conference on Emerging Network Intelligence EMERGING 2013 Copyright (c) IARIA, 2013. ISBN: 978-1-61208-292-9
7. http://cloudtweaks.com/2012/03/kpis-for-cloud-service-providers-customers/ "Key Performance Indicators'.
8. Saurabh Kumar Garg a, ∗, Steve Versteeg b, Rajkumar Buyyaa 'A framework for ranking of cloud computing services'. Future Generation Computer Systems 29 (2013) 1012–1023 journal homepage: www.elsevier.com/locate/fgcs.
9. Calheiros R, Nadjaran Toosi A, Vecchiola C, Buyya R. A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*. Elsevier Science. 2012, Vol. 28, Issue 8.