Indian Journal of Science and Technology, *Vol 9(21), DOI: 10.17485/ijst/2016/v9i21/95287, June 2016* *ISSN (Print) : 0974-6846*
ISSN (Online) : 0974-5645

Federated Architecture for Ranking the Services in Cloud Computing

L. Aruna[1*] and M. Aramudhan[2]

[1]Periyar University, Salem - 636011, TamilNadu, India; arunaloga@gmail.com
[2]Department of IT, PKIET, Karaikal - 609603, Puducherry, India; aranagai@yahoo.co.in

Abstract

Objectives: To improve Quality of Services (QoS) and ranking cloud service providers in federated cloud environment. It is also aimed to resolve various issues faced by the user and providers in cloud. **Methods:** In order to avoid the workload of existing cloud model and to compete with the well-known cloud service providers, customized federated cloud provider architecture was suggested. It is suggested to manage and to maintain QoS for the submitted tasks. The proposed architecture consists of external world, middle world and internal world. Each world plays vital role for customizing the request in federated environment to enhance QoS. To avoid starvation in the architecture to suggest the differentiated module and resolves the various key issues using the Stochastic Markov process model and also evaluating the cloud providers based on the quality of service requirements. **Findings:** Service Level Agreement (SLA) in single and federated cloud plays a vital role in enhancing Quality of Services. Single cloud service provider representation does not provide the skilled QoS, when workload becomes high. Customized federated cloud architecture reduces the drawbacks of single cloud service provider. The proposed architecture was implemented in CloudSim using Java. The simulation result of the proposed architecture proves enhanced QoS to the user and cloud service providers. The proposed federated cloud model enhances the Quality of Services by more than 18% of existing single and federated cloud model. The parameters considered for the simulation for numbers of users, providers, load factor, turnaround time and average load deviation of tasks. **Applications:** This architecture can be used to rank different cloud service providers and it can be trusted for any distributed cloud services with extended QoS.

Keywords: Cloud Sim, QoS and Federated Architecture, Stochastic Markov, SLA

1. Introduction

Cloud computing is a innovative paradigm with in the field of knowledge based Technology that provide cost effective value through the sharing basis in dynamic service surroundings[1]. Service Level Agreement (SLA) is an agreement between the uses of cloud services through the document based that contains the extent of performance promise created by a supplier at the user aspect[2]. Each one Cloud service provider model does not give adept the service, once the work become a high level. In small scale organizations cannot deal with the distinguished cloud service providers or suppliers although they need sufficient various resources are expected by a user or client. And also to beat the grim load balancing of single cloud model and via with the various distinguished clouds service suppliers or providers in federate cloud design was suggested. A Federate cloud could be a federation of a single cloud model that helps to be resolve the excellence contribution within the single cloud service model. In SLA concept is enforced between the cloud user or member and the cloud service providers or cloud service suppliers for economical process of federated cloud architecture was implemented. The SLA parameters are includes memory connected data like performance, memory size, CPU speed, CPU size etc. and also the other important functional SLA parameters are also consider in terms of privacy, security, time interval, cost and execution time etc[3]

The various cloud user needs to represent the each

** Author for correspondence*

persistent and other functional parameters in SLA are also verified. The Cloud service provider of each and every one to chosen for execution of processing and output, must satisfy both the functional and other important non-functional SLA requirements given by a user. The outcome of the best source provisioning to the various[4] cloud user's requirements is one of the not easy tasks in the existing federated cloud method or architecture[5].

In the proposed model, the above parameters are considered in to Cloud Broker Manager (CBM) plays a main role for the liability of various resource provisioning in cloud federated architecture. Each cloud provider like the real time service providers has some broker that interconnects through with the CBM. There are two types of Cloud Broker Manager(CBM) namely $_F$CBM and sCBM.

The roles of $_F$CBM in the proposed cloud customized federated architecture that supports 1. To identify the various category of user in cloud service. 2. To managing the concern user request and the processing the request and also send to next level. 3. In this regard to allocating the available cloud service provider based on the concern user request. The Differentiated Service Module (DSM) is suggested or recommended to cloud service provider for the next level process in $_F$CBM that classifies various requests are consider for the request of various incoming users as either SLA or other non-SLA members in a given architecture. In the implementation part of the first cloud broker manager like the $_F$CBM federated architecture to maintain a service request queue in order to maintain the one or more service queues for users at application level and hence to promote in a differential treatment. It is very scalable, simple, portable, reusable, extensible, flexible and accountable. The DSM cloud service module leads to avoid starvation in many times; hence the proposed ranking method is used to manage the various level of incoming request of users are maintained in a list and also to improve the responses time of the consumed user. To reduce the overload of user requests of $_F$CBM, a secondary CBM method like (sCBM) is proposed to take up the task of resource provisioning available only for the SLA members.

The various important functionality of the secondary method like the secondary cloud broker manager sCBM are also to 1. Identifying the important various available cloud service providers to matching with the concern user requirements, 2. In this area based on the user request and meet out available providers to implement the ranking to the selected service providers and also assigning them for utilizing the services and operations, the next steps is going to finding the best available cloud service provider based on the service register to support and satisfies the user requirements and finally to verify the service performance with respect to the Quality of Service (QoS) like parameters it is approved by the cloud consortium such as flexibility, readability, functionality, adaptability, scalability and reliability. In this chapter, all the above challenges are pointed out in addressed and taken in to solutions are provided. Today, the number of various real time cloud service provider offering lot of services based on demand. Cloud services is increasing quickly and also most of the organization switcher to cloud. The manual process are identify the cloud selection process becomes quite complicated and time consuming[6]. Hence, a stochastic Markov process model is used in secondary sCBM for selecting the corresponding various available cloud providers based on the SLA and user requirements. The important another challenge in secondary cloud sCBM is addressed using the impression of approaches in cloud ranking algorithm. Cloud ranking model is a activated method of assigning ranks to the cloud service providers[7] discovered by Markov process. Based upon the various categories of output in the cloud ranking, sCBM selects the best provider and choose the correct broker from the interconnected available list, and assigns a user tasks to the selected service provider. The functional SLA parameter are includes the various memory related information like memory size and CPU size etc. The SLA functional parameters are consists of performance, cost, execution time, security etc. User requirements are also included for both functional and non-functional level of SLA parameters.

This chapter describes how the SLA user agreement is effectively designed for users and resourceful interoperability is achieved through broker architecture. This chapter also discusses the various aspects like how the customers are linked to the best available or utilized the various cloud service providers through brokers.

2. Proposed Architecture for Federated Cloud

The Proposed cloud architecture consists of three world's namely external world, middle world and internal world. External world includes various users from different locations. Middle world consists of First or primary Cloud Broker Manager ($_F$CBM), secondary Cloud Broker Manager (sCBM), brokers and cloud pool manager. Internal world deals with a number of cloud service providers. Providers may be in different locations. The customized federated cloud architecture shown in Figure 1 demonstrates the fundamental processing sequence of transferring cloud services between users and service providers. Middle world proposed in the cloud architecture consist of three folds.

- Cloud Broker Manager
- Cloud Pool Manager
- Broker Architecture

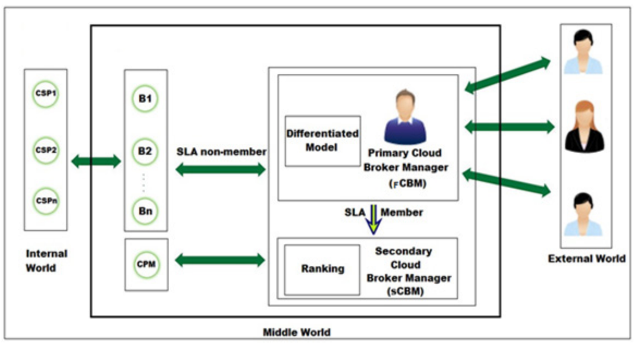

Figure 1. Customized federated cloud architecture.

The customized federated architecture consists of the external world, middle world and internal world. Each one has to plays vital role for customizing the user request in federated environment. SLA management life cycle is overall responsible for resource provisioning in the federated architecture. To avoid starvation in the architecture to suggested the differentiated module and resolves the various key issues. The functionalities of each modules contains as follows.

2.1 Cloud Broker Manager (CBM)

Cloud Broker Manager (CBM) acts as a mediator between users and cloud providers. There are two types of CBM namely FCBM and sCBM. Users send their service request to FCBM. The jobs of FCBM are to identify the type of user, selecting the appropriate broker for the service request, managing the service request and forwarding the service request to provider. FCBM invokes the differentiated service module that identifies the category of the inward user belongs to either SLA or non-SLA member with the help of the information available in Cloud Pool Manager (CPM). CPM is a cloud database. The functionality of CPM is discussed in section 2.2 SLA non-members request are not considered for processing until there is any request belongs to SLA member. Instead of using this strict differential treatment, the proposed approach is used for differentiating, managing and assigning service provider for the user requests. If the incoming service request belongs to SLA member category, the service request is transferred to sCBM. The duties of sCBM are selecting, ranking and choosing the best cloud provider. The stochastic Markov process model is used for selecting the matched cloud providers and the following procedure is used for ranking the selected cloud providers and assigns the service request to the top ranked cloud provider.

It is given by

$$CR(A) = (1-d) + d\,(CR(T1)/C(T1) + \ldots + CR(Tn)/C(Tn))$$

Where,

- CR(A) is the CloudRank of services A,
- CR(Ti) is the CloudRank of services Ti which link to services A,
- C(Ti) is the number of outbound links on services Ti and
- d refers to damping factor, the value can be assigned between 1 and 0.

So, first of all, existing CloudRank does not rank cloud services as a whole, but is determined for each services individually. The proposed CloudRank of services A is recursively defined by the CloudRank of those services which link to services A. The CloudRank of services Ti which link to services A does not influence the CloudRank of services A uniformly. Within the CloudRank algorithm, the CloudRank of a services T is always weighted by the number of outbound links C(T) on services T. This means that the more outbound links a services T has, the less will services A benefit from a link to it on services T. The weighted CloudRank of services Ti is then summed together. The outcome of this is that an additional inbound link for services A will always increase services A's CloudRank. Finally, the sum of the weighted CloudRank of all services Ti is multiplied with a damping factor 'd'. The value for 'd' can be assigned between 1 and

0. Thereby, the extend of CloudRank benefit for a services by another services linking to it is reduced.

2.2 Cloud Pool Manager (CPM)

CPM is a cloud database that has four segments namely cloud table, member details, broker details and provider details. CPM accumulates particulars about SLA and non-SLA members in member details segment. Information about brokers, connection with cloud providers are described in broker details segment. Provider details segment contains name of providers and type of services to be provided. CBM collects the details of members from CPM. Resource availability of a service provider is also obtained using cloud table. Cloud Table in CPM holds the current status of the various cloud service providers in the form of hash Table based on the linked list structure. The linked list structure used in cloud table are clearly shows the volume and status of resources. For example, service provider one consists of virtual machines and each virtual machine is occupied by a user. Service provider n contains virtual machines but only virtual machine two is occupied by a user and other virtual machines are open. FCBM and sCBM uses the cloud table to get the status of service providers.

2.3 Broker Architecture (BA)

Brokers receive the request from users and perform services based on functional and non- functional parameters. The main objective of a broker is to support the users to acquire optimum service utilization of service provider[8]. Broker architecture involves three phases namely Discovery, Allocation & Monitoring and Marketing.

2.4 Discovery

Discovery phase maps a member request with a provider through the components SLA manager, Mapping manager and CPM interface. SLA manager module manages the negotiation between members and service providers[8]. CPM interface gives the information about members and service providers to SLA Manager using CPM. Mapping Manager takes up the responsibility of mapping process between members request and providers parameters and returns the result to the broker manager.

2.5 Allocation and Monitoring

Members request are scheduled and monitored using Allocation and monitoring phase. Scheduler helps to allocate a member task to the selected service provider. Execution process is scrutinized through monitoring manager. If any contravention in the SLA, monitoring manager immediately stops the execution process and sends a violation message to the violation manager. Violation manager examines the type of violation occurred in the process and sends a termination message to the CBM. Service providers are charged based on the type of violation occurred in the execution process.

2.6 Marketing

Cost calculation of a federated cloud is done with the help of marketing phase. Accounting module pools the information about service usage time, type of service obtained and member type. Based on the accounting module billing module calculates the service usage cost and violation cost (if violation occurred)

3. Experimental Results and Discussions

The average execution time of members is highly significant to estimate the performance of service providers. To prove that, presentation evaluation of the proposed architecture is implemented in CloudSim[9] using Java. The general evaluation parameters considered for the experiment are number of users, number of cloud service providers, load factor of cloud service providers, average load deviation, deadline of tasks etc.

The execution time for each task is assigned randomly between 0.1 ms to 0.5 ms. Number of users considered for the experiment are 1000, 5000 and 10000. Number of service providers available is fixed as 10, and deadline for each request is fixed as 0.5 ms.

Load factor is defined as the ratio between the number of requests in CBM and the number of cloud service providers. Load cause that varies dynamically depends on the number of requests arrived at CBM. The average load deviation is defined as the average difference between the load expected and load assigned to cloud service provider. Presentation of cloud service provider decreases when load deviation gets increased. Every cloud service provider consider for the experiment has 50 computing hosts, 10 GB of memory, 2 TB of storage, one processor with 1000 MIPS of ability and a time-shared VM scheduler[10]. Cloud broker on behalf of user request consist of 256 MB of memory, 1 GB of storage, 1 CPU, and time- shared Cloudlet scheduler. The various broker requests instantaneous of 25 VMs and associates one Cloudlet to each VM to be executed.

The investigational results prove that the proposed architecture keeps the load variation in control and provides better presentation of user workload. In experiment a set of 1000, 5000 and 10000 requests are submitted at a time and time taken for identifying the category is negligible. Average execution time for each request is randomly assigned. The average execution time of submitted requests for both SLA and non- SLA members using Strict listed in Table 1.

The results prove that, if the load deviation increases, the total execution time also increases proportionally. This also proves that if the deviation increases the performance is decreased is shown in the following Table 2. In order to maintain the deviation among cloud service providers, load balancing algorithm is required[11]. This requirement is considered as a future work of the proposed model.

The above simulation result reveals that the proposed federated cloud model reduces the average turn- around time by more than 18% than the existing federated cloud model.

Average Load deviation of 1000 users = 52/10 = 5.2
Average Load deviation of 5000 users = 360/10 = 36
Average Load deviation of 10000 users = 300/10 = 30
Standard Deviation of average load deviation of 1000 users = 52. = 2.2803
Standard Deviation of average load deviation of 5000 users = 36 = 6
Standard Deviation of average load deviation of 10000 users = 30 = 5.4772

This experiment has been executed repeatedly for 10 times. After 10 iterations, the load deviation falls between 1.8 to 3.5 for 1000 users, 5.4 to 7.6 for 5000 users, and 4.8 to 6.4 for 10000 users. In summary, according to the author's best knowledge for implementing algorithm, our work is fit for define the all key issues related to identify the cloud providers to offering the various cloud services with QoS, performance, cost and security. The attributes of SMI is also suggested by CSMIC. By implemented this technique can provide how they perform compared to their challenger and therefore they can improve their cloud services.

4. Conclusion

In this paper, broker based architecture is suggested for federated cloud to manage and maintain QoS for the submitted tasks. The advantage of this architecture is extensibility, scalability and reliability. Security can be also extended in this architecture. Decentralized approach is adopted and supervised learning mechanism at broker level can be prompted later in this architecture. Through this architecture, differentiated module, ranking mechanism components and different methodologies can be extended and promote QoS.

Table 1. Average execution time taken using differential treatment

Number of Users			SDM	Proposed Approach
1000	SLA Member	400	120 ms – 200 ms	100 ms – 180 ms
	SLA Non-member	600	300 ms – 600 ms	260 ms – 560 ms
5000	SLA Member	3000	900 ms – 1300 ms	800 ms – 1200 ms
	SLA Non-member	2000	600 ms – 900 ms	500 ms – 800 ms
10000	SLA Member	6000	1800 ms – 2600 ms	1600 ms – 2400 ms
	SLA Non-member	4000	1200 ms – 1800 ms	1100 ms – 1600 ms

Table 2. Shows the Load deviation of cloud service provider

Number of SLA users	Number of request	CSP 1	CSP 2	CSP 3	CSP 4	CSP 5	CSP 6	CSP 7	CSP 8	CSP 9	CSP10
400	Load Factor (40)	36	28	46	39	42	47	46	42	37	31
	Deviation	-4	-12	6	-1	2	7	6	2	-3	-9
3000	Load Factor (300)	280	320	340	270	260	330	350	220	340	290
	Deviation	-20	20	40	-30	-40	30	50	-80	40	-10
6000	Load Factor (600)	560	610	550	630	580	640	620	570	650	590
	Deviation	-40	10	-50	30	-20	40	20	-30	50	-10

5. References

1. Architectural Strategies for Cloud Computing. 2011. Available from: http://www.oracle.com/technology/architect/entarch/pdf/architectural_strate gies _ for cloud _com puting.pdf
2. Kresimir P, Zeljko H. Cloud computing security issues and challenges. Proceedings of 3rd International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies and Services; 2010.
3. Processing visualizing and understanding data from high performance computing. 2014. Available from: http://www.computer.org/web/comp uting,now/archive/may2014
4. Aruna L, Aramudhan M. Fundamentals of cloud computing. Delhi, India: ABC Press; 2013. p. 533-40.
5. Scarfone K, Singhal A, Winograd T. 2007. Guide to Secure Web Services. 2010. Available from: http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf
6. Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications. 2011 Jan; 34(1):1-11.
7. Cloud computing use case discussion group and cloud computing use cases Version 3.0. 2010. Available from: www.cloud-council.or g/cloud_computing_use_cases_whitepaper-4.0
8. Cloud Computing. 2010. Available from: http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment
9. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM. 2010 Apr; 53(4):50-8.
10. Aruna L, Aramudhan M. A novel survey on SLA based load leveling in cloud computing. International Journal of Research in Computer and Communication Technology. 2014 Jun; 6(3):658-62.
11. Prakash A, Chandrasekhar C. An optimized multiple semi-hidden markov model for credit card fraud detection. Indian Journal of Science and Technology. 2015 Jan; 8(2):165-71.