SLA BASED BROKER ARCHITECTURE FOR CLOUD

C.S.Rajarajeswari,
Department of Computer Science, IGCAS, Puducherry

Abstract- Cloud computing is a promising technology in today's IT market. Cloud environment provides numerous number of resources ~~but~~ with less and inefficient interoperability. This leads to the demanding problem to cloud users when access the cloud resources from various service providers and find out the most optimal resource provider who satisfies their QoS requirements. Latest research provides various methods of interoperability management using broker architecture that guides the users to access the resources with fast interoperability .The broker helps the customers to select the best service providers based on their requirements and agreement with service providers. Priority is assigned to each customer based on the SLA agreement with service provider. Different level of service (DiffServ) is assigned to each customer based on the SLA signed with cloud service providers to achieve QoS requirements. This paper describes how Service level Agreement management effectively designed for customer and interoperability are achieved through broker architecture.

Index-Terms: Broker Architecture, differentiated service, SLA member, Service level agreement management and negotiation.

I. Introduction

Cloud computing is a growing valuable technology in Information Technology field. Customers are charged for resource utilization based on "pay-as-you-go" model. Users get their resources from different vendors based on their QoS requirements [8]. But customers faced different technical issues to select resources from different service providers [1] (Amazon, Google Apps). Customers need an intermediary tool between the user and the service provider to handle the difficulties. Difficulties of customers are rectified through "Broker".

User involvement in cloud selection will force cloud service providers to compete each other in providing better and fast services to users. To solve this cloud brokers could be used which can serve as an intermediary between cloud service provider and user [3].

Customers get information about cloud service providers through broker. Broker manages the negotiation, resource management and termination process between customer and resource provider. All the negotiation and termination processes are handled by the broker. Cloud broker minimizes the time, cost and work load of cloud customers as well as cloud service providers.

Brokers acts as an intermediary between customers and providers. Brokers take care of customer's requirement and send it to the service provider. Brokers deals with the QoS requirement of customers, Service level agreement between customers and providers, negotiation of SLA's, monitoring SLA violation, terminates connection and so on [2]. Here customers can liberally use the resources of providers or transfer from one cloud platform to another platform.

Service Level agreement is an important issue between customers and providers. SLA deals with various QoS parameters like functional (CPU cores, Memory size, CPU speed, storage size) and non-functional (Response time, Delay, Budget, Deadline, Data transfer time) requirements. Different Level of service is assigned to the users depend on the level of SLA opted by the user based on their requirements. Broker architecture maintains two SLA service queues for users at application level in SLA module and accordingly promotes their differential treatment. It is simple, extensible, flexible and portable. This paper discusses how the customers are linked to the better available cloud service providers through brokers. Brokers plays a vital role in inter cloud processing. Differential service Module is implemented in SLA negotiation

The rest of this paper is divided as follows. Section 2 we discuss previous works related to cloud service brokering. Section 3 explains cloud service broker architecture and various services provided by cloud broker. Section 4 presents Sla management process. Section 5 concludes the paper with summary and future work of this paper.

II. Related Work

The main aim of inter cloud service brokering in cloud has received a lot of attention in academic and industry in the recent years. Broker based inter cloud processing is an emerging concept in cloud based services. This connects a set of technologies, protocols and languages to communicate between customers and providers.

(Buyya et al., 2010) presented the architecture of a Cloud computing environment named Inter-Cloud to support the applications across multiple vendor Clouds. The architecture consists of three components. They are Cloud Broker, Cloud Coordinator and Cloud Exchange. A user submits the request to Cloud broker and the broker sends the requirement to Cloud Coordinator. The coordinator acts as a doorway between cloud providers and brokers. Cloud exchange acts as a mediator between cloud user and service provider.

(Metsch et al., 2010) implemented prototype broker architecture, based on a combination of the core SLA@SOI framework and the RESERVOIR1 framework. This method allows an easy and on-demand provisioning of virtual infrastructure resources within a heterogeneous cloud platform.

(Kertesz et al., 2011) investigated the use of autonomic computing principles for resource management and SLA enforcement in Cloud environments. The proposed architecture consists of three main components: a Meta-Negotiator, Meta broker and Automatic service deployer. Meta-Negotiator manages the agreement negotiations, Meta broker responsible for selecting the proper execution environment and an Automatic Service Deployer responsible for service virtualization and on-demand deployment.

The proposed architecture combines the ideas of the above mentioned methods. It mainly deals with SLA management, SLA negotiation between customer and resource provider and SLA agreement between user and provider.

III. CLOUD SERVICE BROKER ARCHITECTURE

The following diagram (Figure 1) illustrates the basic function of cloud service between customer and service provider.

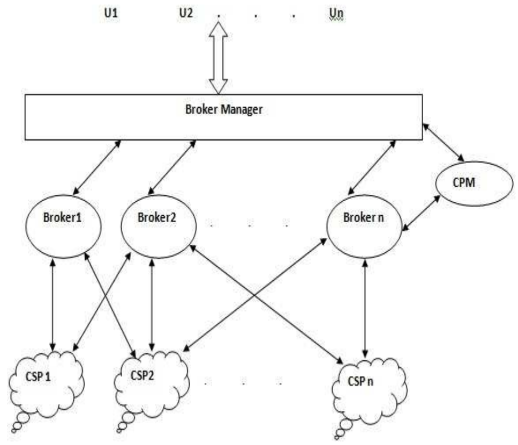

Figure 1. Cloud Architecture

Broker Manager (BM) acts as a mediator between user and cloud broker. BM gets a user request and sends the request to a cloud broker [12]. The functional architecture of BM is depicted in figure 2.

BM Handler in figure 2 used to interact with other components of BM architecture as well as with users. Data base stores the details of the execution process. Match maker assist the BM to find a best cloud service provider who satisfies a user request in an efficient way. Status in BM gives the information about the current execution.

Cloud Pool Manager (CPM) is a cloud database which stores the details of brokers, members and service providers. In some cases BM interacts with CPM to get the details of members, service providers or brokers.

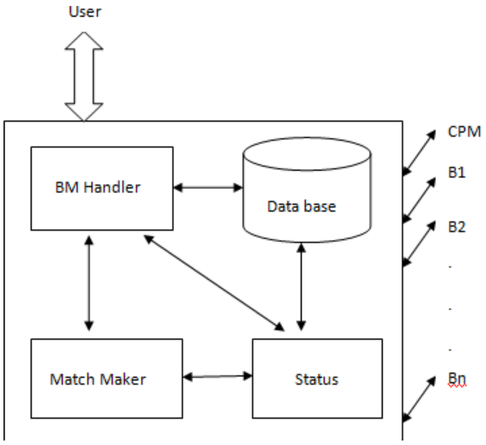

Figure 2.Broker Manager Architecture

Users involved in cloud management are cloud members. Memdet in CPM stores all the cloud member details and SLA member details.

When a user enters into a cloud for the first time, they are not the SLA members of the cloud. After the execution process gets over between the cloud user and the cloud service provider, the Broker Manager sends a membership request message to the user. If the user satisfied with the performance given by the cloud service provider, then the user accepts the membership request message and sends an acknowledgement message to the Broker Manager. After receiving the acknowledgement message from the user, Broker Manager creates and assigns a slamemid to the user and sends it to the user for future correspondence. Broker Manager sends the created slamemid to the Cloud Pool Manager and stores the id values into the database.

SLA members are separated into two categories. They are High SLA member and Low SLA member. High and Low are assigned for the SLA members based on the number of successfully processed executions.

Figure 3 shows the functional architecture of cloud broker.

Discovery

SLA manager in Discovery manages the negotiation process between cloud member and service provider. CPM interface gives the details of the cloud members or SLA members. Mapping Manager finds a best cloud service provider who satisfies a user requirement.

Allocation & Monitoring

Scheduler helps to schedule a user task to the selected service provider. Monitoring manager monitors the execution process between user and service provider. During execution if any violation occurs then the monitoring manager immediately sends a violation message to the Violation manager. Violation manager immediately stops the execution process and sends a type of violation occurred as well as the termination message to the Broker manager.

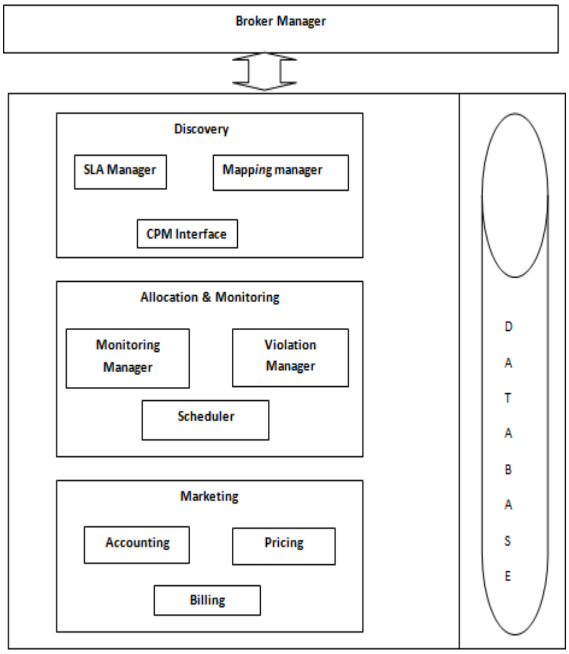

Figure 3.Broker architecture

Marketing

Accounting module stores the service usage information of resources. Pricing determines the amount based on the resource utilization. Billing [2] calculates the usage cost as well as violation cost (if violation occurred).

Database

Database in broker architecture maintains the status of discovery, allocation & monitoring and marketing modules. Database used to get the details at the time of recovery when cloud broker fails to do the execution.

Figure 4 shows a general services provided by a cloud broker.

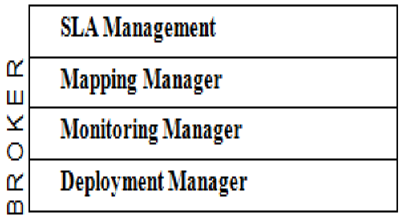

Figure 4. Broker services

As depicted in Figure 4 cloud service broker acts as a mediator between customers and cloud service providers. Cloud service providers supply attractive services to the customers. The main objective of the cloud broker is that they assist the users to the optimum utilization of the resource offered by the providers.

IV. SLA MANAGEMENT

Service Level Agreement (SLA) is defined as a formal agreement between cloud service providers and customers. Flexible and reliable management of SLA agreement is of ultimate importance for both cloud providers and consumers [4][7]. The general function of the SLA management is divided into three parts. They are

✓ SLA Negotiation
✓ SLA Monitoring
✓ SLA Termination

In this layer broker acts on behalf of the customer to execute all the SLA management processes. Broker performs many services to the customers including both functional and non-functional activities. In this paper functional activity consists of CPU speed, CPU cores, Memory size, and Storage size. In the same way non-functional activity consists of cost, deadline, response time and data transfer time.

Table 1 shows the functional and non-functional activities performed by broker.

Functional	Non-functional
CPU speed	Cost
CPU cores	Deadline
Memory size	Response time
Storage size	Data transfer time

Table 1 - SLA parameters

A. SLA Negotiation

In this layer a negotiation process is started by the cloud customer with the help of the broker in order to reach an SLA agreement between customers and cloud service providers. User asks the broker to provide the template of the service providers. Based on the template received from the broker the user submits the request to the SLA manager. SLA manager send the request to the mapping manager.

Customer sends a request message to the SLA manager (1). SLA manager send the request to the mapping manager (2). Mapping manager first parse the user request and send the information to the monitoring manager (3). Monitoring manager sends a parsed SLA agreement to the resource provider. Deployment manager (4, 5) forwards the message to both inter cloud gateway and cloud platform. From the result of the Inter cloud platform and cloud gateway (6), the monitoring manager (7) sends an acknowledgement agreement message to the mapping manager (8). Mapping manager sends a result to the SLA manager (9) and to the user (10).

Based on the acknowledgement agreement message received from cloud broker, cloud user wil select the best cloud service provider who satisfies their QoS requirement. After selecting the provider, cloud user sends a resource request message to the provider.

B. SLA Monitoring

Based on the agreement between customer and service provider resources are provided to customers. Resources are provided to customers based upon the resource request given by a customer. Resource request is passed from the user to the cloud provider through SLA Manger, Monitoring manager, Mapping manager and Inter cloud gateway. During the resource management process a broker verifies whether any SLA violation occurs between customer and service provider. If any SLA violation occurs between them then the broker sends a SLA violation message to the service provider and the transaction dropped immediately between them.

C. SLA Termination

After completing the execution, customer sends a SLA termination request message to the cloud service provider. SLA termination request message from the cloud customer is passed to the service provider through SLA manager (1), Mapping manager (2), Monitoring manager (3) and Deployment manager (5, 6). Based on the termination request given by the customer, cloud service provider terminates the communication process immediately and sends an acknowledgement termination message to the customer. Acknowledgement termination message from cloud service provider to customer has been passed through inter cloud gateway, monitoring manager, mapping manager and SLA manager.

After terminating the current execution customers may contact the same cloud service provider or some other cloud service providers for their next request through cloud broker using the SLA negotiation method as we discussed. Hence the proposed cloud broker architecture provides a clear idea about the activities of cloud broker and SLA management between user and provider. SLA management (SLA negotiation, SLA monitoring and SLA termination) between cloud user and cloud service provider is a significant process. Cloud broker executes this process in an efficient way.

With the help of a cloud broker cloud customer get the effective utilization of resources given by the cloud service provider without spending more waiting time. Cloud broker reduces the time as well cost of both customer and cloud service provider.

V. CONCLUSION

This paper described the broker architecture in cloud computing environment. Broker Architecture provides a optimum resource utilization than compared with the traditional cloud computing method. It also minimizes the waiting time of the user. All the negotiation and termination processes are handled by the broker. Cloud broker minimizes the time, cost and work load of customers as well as service providers.

Each cloud providers are provided with a different infrastructure offer, cloud interfaces and price policy for the customers. Cloud brokering mechanism helps to avail optimum performance across multiple cloud providers according to the user demands and manages the resulting infrastructure. Brokering mechanism provides with scheduler that suggests optimal selection of resources across multiple cloud providers.

Brokers acts as an intermediary between customers and providers. Brokers take care of customer’s requirement and send it to the service provider. Brokers deals with the QoS requirement of customers, Service level agreement between customers and providers, negotiation of SLA’s, monitoring SLA violation, terminate connection and so on. Here customers can liberally use the resources of providers or transfer from one cloud platform to another platform.

The broker should be fully operational and its interface should be known in advance to the cloud service provider and cloud customers. Brokers may have several links with different cloud service providers. Sometimes they all work in a same time, this will cause some problem.

Cloud service providers may say that the problem is somewhere other than with them. Hence cloud brokers should reduce these limitations by working with cloud service provider.

In this paper, the SLA-members are stored in the database with free of cost in order to improve the cloud technology, but it is not possible in all the cases. So in future, convert the free cost of SLA member into paid services in order to avoid the heavy load of broker manager. If all the incoming users are SLA members, then the non SLA member will not get the chance of resource utilization. It will create starvation. In future we extend this work to use simulation tool to verify and evaluate the efficiency performance of various SLA match making algorithms.

REFERENCES

[1] Bernstein .D. et al(2009), “blueprint for the Intercloud: Protocols and formats for Cloud Computing Interoperability”. In Proceedings of 4th International Conference on Internet and Web applications and Services.

[2] Buyya R., Ranjan R. and Calheiro R. N. (2010), “InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services”. In Proceedings of *10th* International Conference on Algorithms and Architectures for Parallel Processing.

[3] Gartner (2009), “Cloud Consumers Need

Brokerages to Unlock the Potential of Cloud Services”.

[4] Linlin Wu and Buyya R. (2010), “Service Level Agreement (SLA) in Utility Computing.”

[5] Metsch T., Edmonds A. and Bayon V. (2010), “Using Cloud Standards for Interoperability of Cloud Frameworks.” Tech. Rep. 2010.GICTF (2010). Use Cases and Functional Requirements for Inter- cloud Computing. White Paper 9-8-2010.

[6] Kertesz A., Kecskemeti G. and Brandic I. (2011). “Autonomic SLA-aware Service Virtualization for Distributed Systems.” In Proceedings of 19th Euromicro International Conference on Parallel, Distributed and Network-based processing.

[7] Foued Jrad, Jie Tao and Achim Streit (2012), “Sla based Service Brokering in Inter cloud Environments”. Closer 2012-2nd international conference on cloud computing and services science.

[8] M.Adel Serhani, Rachida Dssouli, Abdelhakim Hafid, Houari Sahraoui (2005), “A QoS Broker based Architecture” .In Proceedings of the IEEE international conference on Web Services.

[9] A. Keller and H. Ludwing, “The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services”, IBM Research Report, May 2002.

[10] W.T. Tsai, R. Paul, Z. Cao, L.Yu, A. Saimi, B. Xiao, “Verification of web services using an enhanced UDDIserver” Eighth IEEE International Workshop on Object- Oriented Real Time Dependable Systems, Guadalajara, Mexico, January 2003.

[11] Theilmann W., Happe J., Kotsokalis C., Edmonds and Kearney K. (2010). Reference Architecture for Multi- Level SLA Management. In Journal of Internet Engineering, Vol 4, No 1, December 2010.Attila Csaba Marosi, Gabor Kecskemeti, Attila Kertesz, Peter Kacsuk (2011) “FCM: An Architecture for Integrating IaaS cloud systems”. In Second International Conference on Cloud Computing 2011.