Journal of Theoretical and Applied Information Technology
20th November 2015. Vol.81. No.2

© 2005 - 2015 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DESIGNING ASPECT AND FUNCTIONALITY ISSUES OF CLOUD BROKERING SERVICE IN CLOUD COMPUTING ENVIRONMENT

[1]SOUVIK PAL, [2]PRASANT KUMAR PATTNAIK
[1]School of Computer Engineering, KIIT University, India
[2]School of Computer Engineering, KIIT University, India
E-mail: [1]souvikpal22@gmail.com, [2]patnaikprasntfcs@kiit.ac.in

ABSTRACT

Cloud brokering service is an intermediate service which enables the producer-consumer business model enforcing the easy access to cloud services from Cloud Service Providers (CSPs). Cloud broker is to provide a platform where broker collects the information from the user, analyze the data, and sends those data to the CSPs. Cloud broker also provides data integration services and modeling the data across all the components or units of the cloud services. This paper deals with designing criteria and issues of cloud broker, system activity of broker, and sequence diagram of system design with implementation procedure.

Keywords: *Cloud computing, cloud broker, Virtualization, Virtual Machine.*

1. INTRODUCTION

Cloud computing is a buzz word in the recent IT industry and as well as academic research, which is strongly concerned with customer involvement with Cloud Service Provider. A Cloud itself an infrastructure or framework that comprises a collection of physical computing resources i.e. a set of hardware, storage, memory, processors, networks and bandwidth, which can be structured into services in agreement with the requirement of the client, that can grow or shrink in real-time scenario [1][2]. Cloud computing, also known as internet computing, provides on-demand services in which shared resources, software, information, software and other application specific services are made available according to the consumer requirement with minimal attempt at specific time[3]. Cloud broker is such a Business Model which acts like an agent which helps the clients to choose the right resources. The traditional IT users are not capable of keeping track of all their activities. So in this situation, the cloud brokers are there to help the clients to track their activities and help to choose the best resources which can be provisioned with minimal effort and less time according to the need of the clients. Cloud brokers are responsible for the governance and the management of the cloud environment [4][5]. Depending on the workload classification, the IT clients will decide to choose the cloud infrastructure; in that case cloud broker will provide the platform, on which the client will get the best sourcing option for a cloud service, provisioned resources, and also a consolidated bill.

The paper is organized as follows:
Section 2 deals with the literature survey of the related work, need for cloud broker and objective of the study. In the section 3, we have discussed about the design aspect of the cloud broker, system activity and also the sequence diagram of the brokering services. In the section 4, we have presented the implementation procedure of the cloud broker using Object Oriented Concept. And at the end of the paper, conclusion section brings to a close of the work.

2. LITERATURE SURVEY OF THE RELATED WORK

Cloud brokerage service is a novel area of research. In this age of intermediation, cloud brokerage service plays a vital role in the Industry. Cloud broker creates an interface to facilitate the IT user to choose the appropriate data center capable of providing adequate resources according to the requirement of the user. In the recent years, experts have focused on brokering services in cloud computing environment.

Grivas, Kumar, and Wache, 2010 [5], have described the changes of business processes in the industry. They have focused on minimization of downtime in business process. They have proposed an approach for change management, which is capable of handling the processes.

Nair, Porwal, et al, 2010 [6], have discussed cloud brokerage and cloud bursting in their paper. Cloud broker allows different cloud providers to provide composite services that would help to grow in the competitive market. They have presented an architectural framework for cloud brokering services and also the security issues concerned to the models described in their paper.

Haresh, Kalady, and Govindan, 2011 [7], have proposed a system where the agents namely resource agent, provider and consumer agent play the vital role in cloud computing environment. Resource agent uses the resource allocation method. They have described the way to get the resources with the least price without knowing Cloud Service Provider (CSP) and the location of CSP.

Song, Bae, Lee, and Youn, 2011 [8], have proposed utility adaptive cloud brokerage means that allows the cloud user to get more flexible services from CSP. They have also described the communication procedure within the mechanism. This mechanism provides cost-effectiveness and also perfect resource-utilization.

Sundareswaram, Squicciarini, and Lin, 2012 [9], have focused on brokerage-based architecture in cloud computing environment. Cloud broker is mainly meant for service selection. They have designed a unique indexing technique which manages the huge number of CSPs. They have also proposed a service selection algorithm that helps the brokerage service to select the services.

Grozev and Buyya, 2012 [10], have described about achieving enhanced Quality of Service (QoS) and reliability. They have focused on cost-efficiency by employing multiple clouds. Their aim is to motivate the issues of Inter-Cloud architectures and also cloud brokerage mechanisms. In their paper, they have also discussed the way to facilitate brokering service to distributed appliance by Inter-Cloud environments.

Ferrer, Hernndez, et al, 2012 [11], On behalf of the cloud consumer, their representatives i.e. brokers are directly accountable for managing the resources, provisioning the resources, and also for scheduling. For business point of view, when different cloud providers collaborate or interconnect their resources or infrastructures to provide a better service allowing the resources to be shared, a Federation can be achieved. Cloud broker is business service that plays on behalf of the customer with the intention of provisioning the resources and deploying the application components.

Jin, Kwok, and Yan, 2013 [12], have explained cloud brokerage by resource multiplexing. They have identified the problem of consuming resource in competitive market via cloud brokerage. They have formulated non-cooperative game modeling to solve the problem. They have also used Nash equilibrium for competitive resource procurements.

Barsoum and Hasan , 2013 [13], have proposed the idea of hiding the sensitive data from CSPs using the enabling indirect mutual trust. Their proposed scheme facilitates the data owner to send the sensitive data to CSP and to perform block-level process on the said data. Their scheme also allows only the authorized users to access those data and they have also discussed security issues and prototype implementation on Amazon.

Caton, Haas, et al., 2014 [14], have presented social compute cloud which enables the infrastructure provisioning through “friend” relationships. Service provider offers the virtualized container which works on smart devices while connected to social network. This paper deals with the resource allocation process with the help of sharing preferences.

Pal and Pattnaik, 2015 [15], have discussed about FCFS algorithm, Johnson sequencing algorithm, queuing model, and waiting time. They have used Johnson sequencing algorithm to minimize the waiting time in the queue and s well as in the system using queuing model with finite capacity and multi-server capability.

Calero and Aguado, 2015 [16], have presented a monitoring architecture concerned to the CSP and cloud user. This architecture allows the user to customize the metrics. The cloud providers can easily track the services used by the users. CSP have used Adaptive distributed monitoring technique which is implemented in cloud infrastructure.

© 2005 - 2015 JATIT & LLS. All rights reserved·

2.1 Need for Cloud Broker

Cloud broker is the core service, which is offered by cloud environment. It facilitates the IT user helping out to search, choose and contract for the services [17]. Through the cloud broker, the clients can easily get the services and also can deploy the applications onto cloud platform. Brokering services can be taken place on request of either external customer request or internal customer request or scaling request or renewal request [17]. Cloud broker, as an intermediation service, first does the identity and access management capabilities. When access is authorized, according to Service Level Agreement (SLA service), all the User data and requirements are reported to the service provider. Cloud broker will contact with the service level management service for contracting the consumer service offer. This process involves the instantiation and the commissioning matter of the preferred service. If the required resources are available, then according to the SLA, the resources are provisioned. And if the resources are not available, the broker will offer for the next service. Cloud brokers are also responsible for incident or event reporting like renewal request or scaling request. And the most important function of cloud broker is supervision on pricing and billing service which is the most serious aspect from business point of view.

2.2. Objective of the study

In the previous section, we have discussed the different issues of brokering service and the requirement of cloud broker. Cloud broker creates an interface to facilitate the IT user to choose the appropriate data center capable of providing adequate resources according to the requirement of the user. In this paper we have discussed modeling and designing aspect of brokering services which includes different functionalities of cloud broker say automatic resources provisioning, scheduling of user request, accounting of renewal request, making Service Level Agreements. In the next phase of the paper we have discussed the system activity and sequence diagram of system design with implementation procedure using Object Oriented Concept.

3. DESIGNING ASPECT OF A CLOUD BROKER

In this section, we will discuss the design aspect of a broker and illustrate a schematic diagram which describes the functionalities of each component of a cloud structure especially cloud broker. Subsequently we have discussed system activity which shows the system flow and respective sequence diagram that describes the different activities between different entities. In this paper, we are mainly focusing on the brokering service, its issues, and implementation procedure using object oriented concept.

3.1. Schematic Diagram of the Cloud Broker

In our design as shown in the figure [1], there are a number of cloud users who make the requests onto the user interface, developed by the Cloud Service Provider. That interface is connected with the user database so that at this point, identity of the new user can easily be created and validated. And the existing users can authenticated them and their access management is controlled by the interface. While the clients get authorized to get services from the service providers, they put request or we can say, events are triggered by the user. The requests may be Resource-based or Infrastructure-based or Platform-based or Software-based or Storage-based. At that point, depending of the types of the requests, there may be multiple numbers of queues. Cloud broker has the capability to classify the requests and make the identification of access requests. Depending on the user request, what kind of request is made by the user, which is reported to the cloud broker as an incident or event reporting. That means that request is accepted though cloud broker program module which is described later. When there comes multiple numbers of requests, cloud brokers needs to apply different scheduling algorithms like Round-Robin Procedure[18][19][22], Dynamic Round-Robin Procedure [20][22], Striping Procedure [18][21][22], Packing Procedure [18][19][21][22] and also apply load-balancing algorithms like Free-CPU-Count based Procedure [18][21][22] and Ratio-based Load Balancing Procedure [18][22] for Scheduling and load balancing of incoming requests. Cloud broker also provides Service Level Agreements (SLA), based on which a relationship between user and cloud broker has been established. Users may also make scaling request or renewal request for the existing applications or for the services. Cloud broker also provides pricing and billing services. Cloud billing concept is on “pay as you go” model that means the cloud users are being charged according to the usage of the cloud infrastructure.

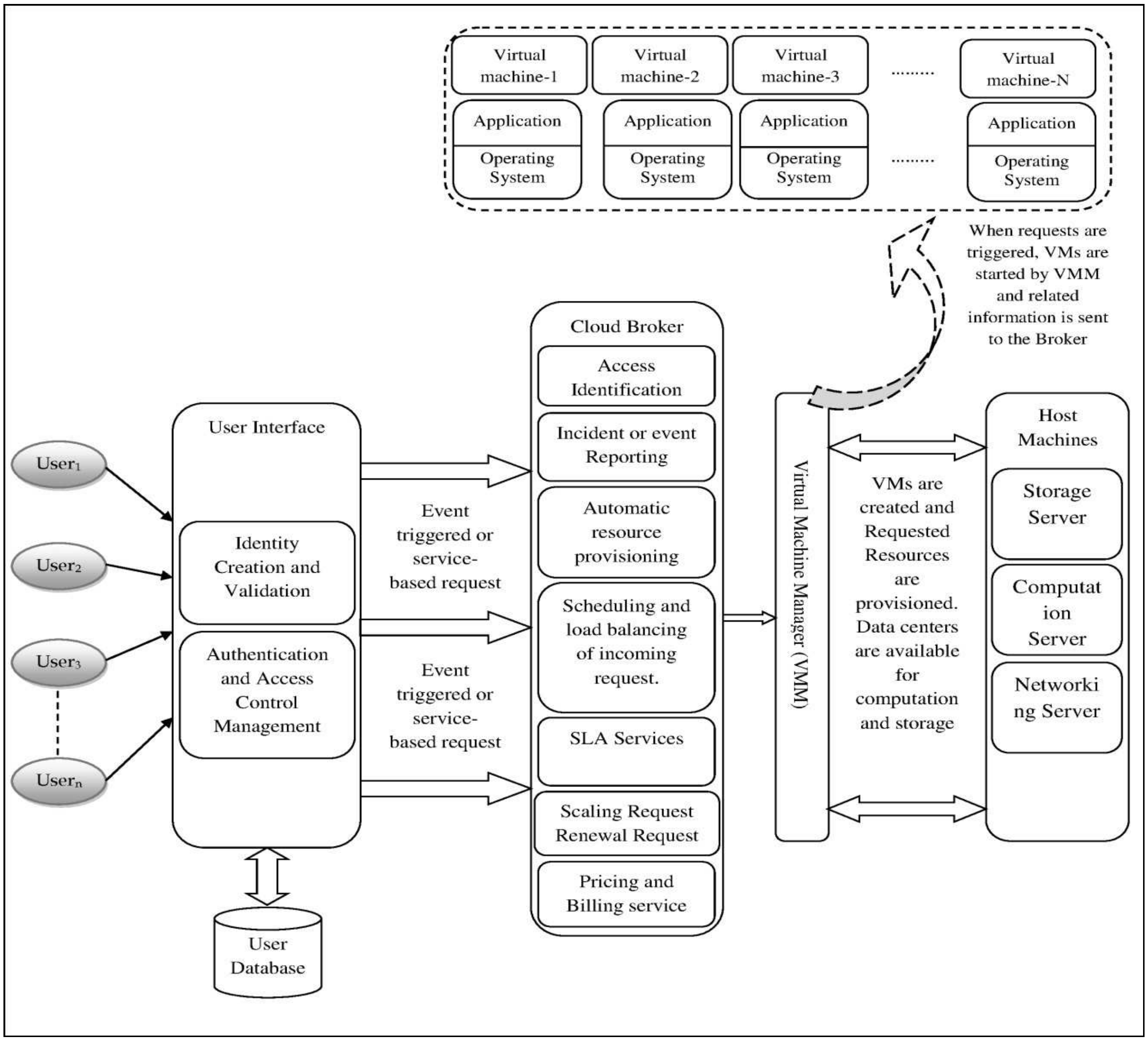

Virtual machine-1
Virtual machine-2
Virtual machine-3
....
Virtual machine-N
Application
Operating System
When requests are triggered, VMs are started by VMM and related information is sent to the Broker
Cloud Broker
Access Identification
Incident or event Reporting
Automatic resource provisioning
Scheduling and load balancing of incoming request.
SLA Services
Scaling Request Renewal Request
Pricing and Billing service
User Interface
Identity Creation and Validation
Authentication and Access Control Management
User1
User2
User3
Usern
User Database
Event triggered or service-based request
Event triggered or service-based request
Virtual Machine Manager (VMM)
VMs are created and Requested Resources are provisioned. Data centers are available for computation and storage
Host Machines
Storage Server
Computat ion Server
Networki ng Server

Figure 1: Design of Cloud Broker

After analyzing and processing the user-request in each modules of the cloud broker, it directs the Virtual Machine Manager (VMM) to create the Virtual Machines (VMs). VMM has the vital responsibility to organize and manage over the accesses of VMs. Virtual Machines are similar to the interface to the clients, which have own operating system on which the users can deploy their applications or may configure it depending upon the requirements. The users can directly interact with VMs. Virtual Machines have the capability to provide a unified and consistent view to the clients, so that clients think they are using only one system but originally VMs facilitate them with aggregation of the computing resources from multiple machines. VMM monitors how the system and computing resources are provisioned for sharing. It also regulates the mapping policies how the Virtual machines are mapped onto host machines so that from the resource pool of the Service Provider, the data, storage and computing resources can easily be accessed and retrieved. When requests are triggered, after analyzed by the broker, VMs are started by VMM and related information is sent to the cloud broker. When VMs are created, the required resources are accordingly provisioned from the host machines. Efficient usage of servers may be capable of maximizing the sharing of systems and computational resources, minimizing the cost complexity, and reducing the waiting time.

3.2. System Activity of Brokering Services

In this section, we will discuss the system activity or the system flow. In the previous section, we have already explained the designing aspect of cloud broker. So, in that designing environment we are going to present how the system activity flows in that section.

In the system flow shown in the figure [2], user interacts with the cloud broker through a user interface which is responsible for user authentication. If the user already exists, then the user authenticated and validated and if not, this interface creates new identification for that particular user and validate. Depending upon the type of the requests, there may be multiple queues, each of which contains similar types of user request. Then the request goes to the broker and the functionalities of the broker are already discussed in the previous section. When the cloud broker analyzes the request and approves it with the availability of the computing and system resources in the cloud infrastructure provided by the Cloud Service Provider, event is triggered. Upon that triggered event, Virtual Machine is started and after providing the services Virtual Machine is stopped. In that meanwhile, the ID of the created VM, the starting time of VM and the finishing time of VM are sent to the cloud broker. If the user wants to renew the existing service, the user again triggers an event for extended service and the same procedure occurs. The ID and the details of the renewal request are sent to the cloud broker. While the request is accomplished that means the service that is requested by the user is completed, the user logged out the system.

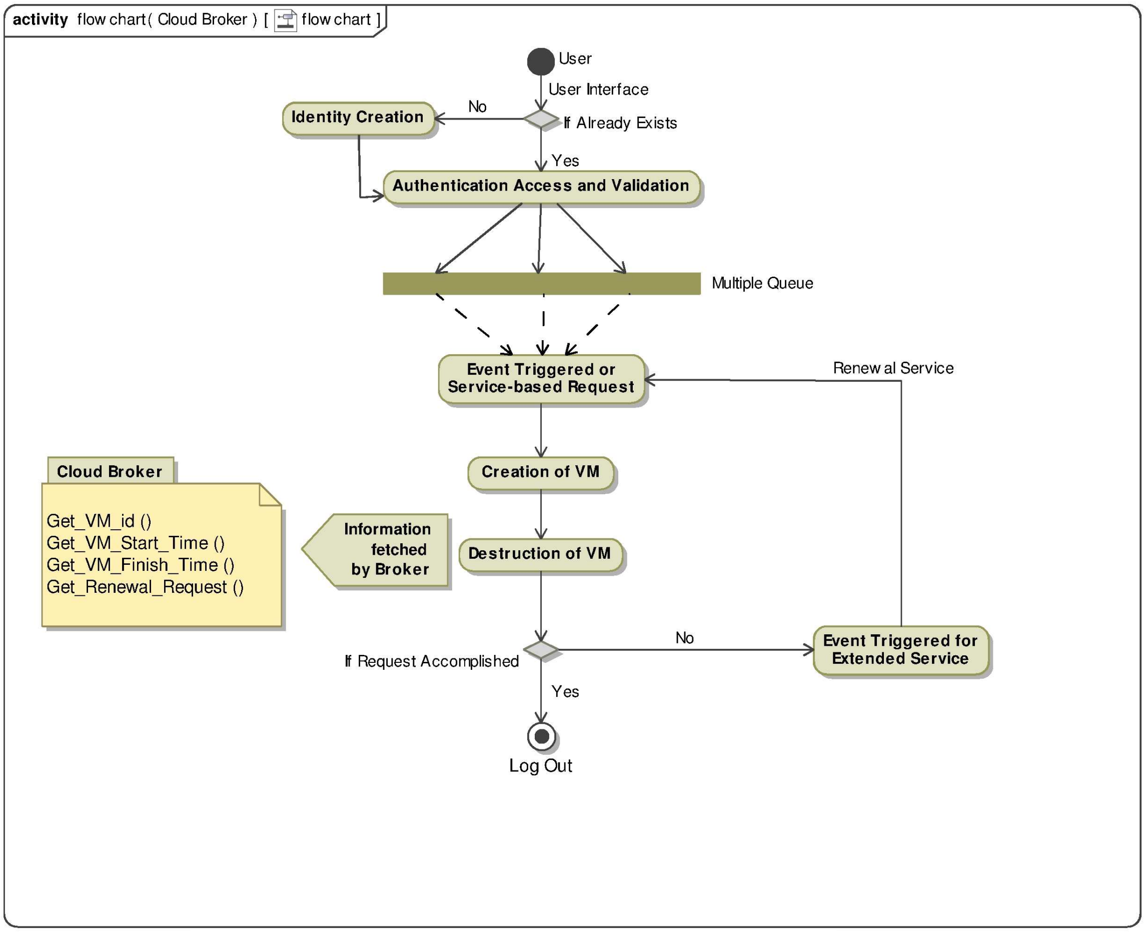


Figure 2: System Flow of Cloud Brokering Services

3.3. Sequence Diagram of the System Design

This section elaborately describes the sequence diagram which explains the control sequence of our design. This sequence diagram helps to understand how the control flows through the different modules shown in the figure [3] below. We will discuss here the sequence diagram step-wise.

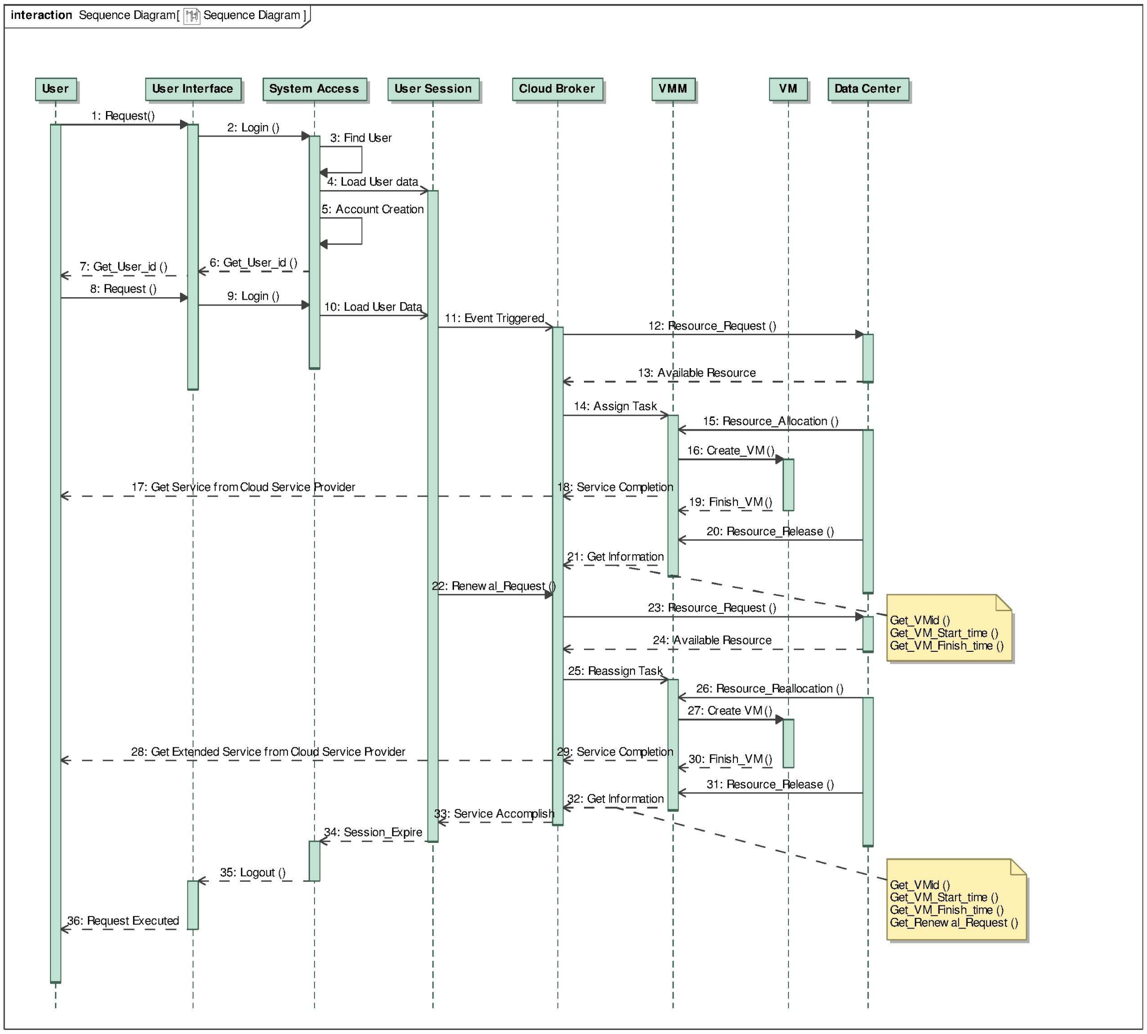


Figure 3: Sequence Diagram of System Design

The steps of the sequence diagram of figure [3] are following below:

Step 1: User first throws a Request () to the User Interface for getting logged in, which contains the user details.

Step 2: User Interface receives the user details and sends those data for system access, involving login() method.

Step 3: System Access module is associated with the user database, where user details are verified and authenticated.

Step 4: After authentication of the user, system loads the user data to the user session.

Step 5: If the user doesn't exit, system will create the account and stores in the user database.

Step 6: System sends this account details to the user interface, which is received by the end user.

Step 7: From the user interface, user can easily get its ID details.

Step 8: After creation of user account, user sends Request() message the User Interface for getting logged in.

Step 9: Receiving the user details, user interface sends those data to the system access for authentication.

Step 10: After authentication of the user, system loads the user data to the user session.

Step 11: Within the user session, when event is triggered by the user, it comes to cloud broker.

Step 12: After getting the triggered event, the cloud broker analyzes the user-request. And it makes a Resource_Request() to the appropriate datacenter for the availability of the resources those are requested by the user.

Step 13: Then that datacenter makes a reply containing the report of availability of the resources.

Step 14: After getting the list available resources, Cloud Broker assigns tasks to the Virtual Machine Manager (VMM).

Step 15: For completion of the assigned tasks, data center allocates the resources to the VMM using Resource_Allocation() method. Different allocation algorithms may be used to allocate the resources.

Step 16: After allocation of the resources, VMM starts the Virtual Machine on which users can easily run or deploy their applications. Each VM is dedicated to a particular user so that the cloud broker can easily monitor the accounts of each user.

Step 17: Cloud user gets the requested service from the CSP.

Step 18: Cloud broker gets the service completion from the VMM. VMM sends the completion message to the cloud broker when the user-requested service has been accomplished.

Step 19: After accomplishing of the service VMM will stop the respective Virtual Machine.

Step 20: The resources on which VM was working, those resources has been released for another service.

Step 21: All the details like respective VM_id, VM_Start_time, VM_Finish_time are send the cloud broker by the VMM so that the cloud broker can easily keep an account of the that VM and respective cloud user.

Step 22: If user wants to renew its request that means wants to get extended service, it sends Renewal_Request() to the cloud broker.

Step 23: When cloud broker gets the renewal request, it again analyzes that request and sends Resource_Request() to the suitable datacenter.

Step 24: The cloud broker gets the list of available resources from the datacenter.

Step 25: Cloud broker reassign the extended task to the VMM.

Step 26: Data center again allocates the required resources to the VMM with the help of Resource_Allocation() method.

Step 27: When allocation of the resources completes, VMM starts the Virtual Machine on which users can use the cloud infrastructure for renewal service and for their applications.

Step 28: Cloud user gets the requested extended service from the CSP.

Step 29: Cloud broker gets the service completion from the VMM.

Step 30: After accomplishing of the service VMM will stop the respective Virtual Machine.

Step 31: After completion of the service, the resources has been released for another service.

Step 32: All the details like VM_id, VM_Start_time, VM_Finish_time, Renewal_Request are send the cloud broker by the VMM.

Step 33: Cloud broker sends the service accomplishment message to the user session.

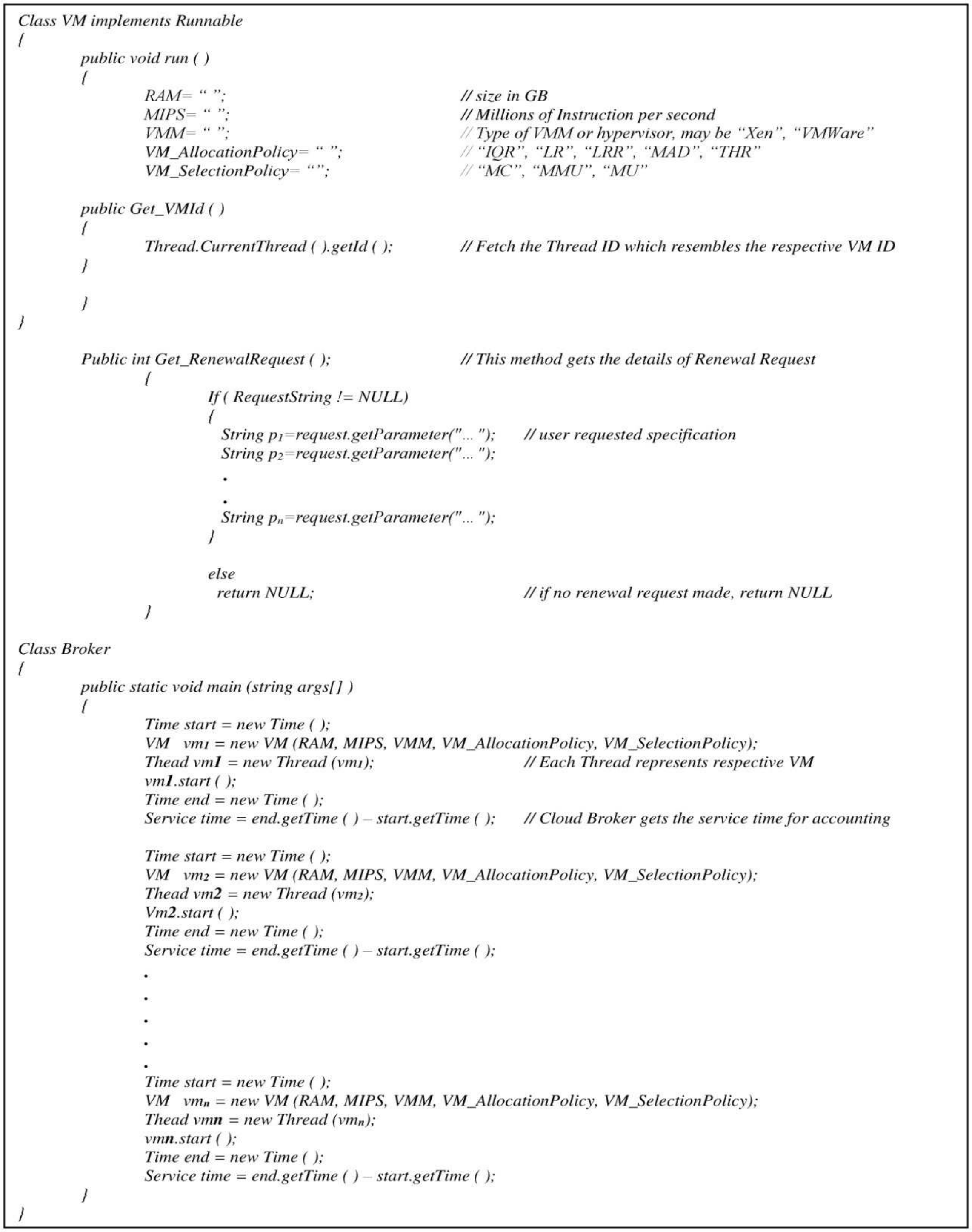
Step 34: While the request has been served, the user session expires and that message is sent to the system.

Step 35: The user can then log out the system after completion.

Step 36: User Interface then sends the request execution completion message to the user.

4. IMPLEMENTATION OF BROKERING SERVICES USING OBJECT ORIENTED CONCEPT

In this section, we will discuss the implementation procedure using Object Oriented Concept. We have used threading concept for implementation purpose shown in the figure [4] below.



```
Class VM implements Runnable
{
          public void run ( )
          {
                    RAM= " ";                                   // size in GB
                    MIPS= " ";                                  // Millions of Instruction per second
                    VMM= " ";                                   // Type of VMM or hypervisor, may be "Xen", "VMWare"
                    VM_AllocationPolicy= " ";                   // "IQR", "LR", "LRR", "MAD", "THR"
                    VM_SelectionPolicy= "";                     // "MC", "MMU", "MU"

          public Get_VMId ( )
          {
                    Thread.CurrentThread ( ).getId ( );         // Fetch the Thread ID which resembles the respective VM ID
          }

          }
}

          Public int Get_RenewalRequest ( );                    // This method gets the details of Renewal Request
                    {
                              If ( RequestString != NULL)
                              {
                                String p1=request.getParameter("... ");      // user requested specification
                                String p2=request.getParameter("... ");
                                .
                                .
                                String pn=request.getParameter("... ");
                              }

                              else
                               return NULL;                              // if no renewal request made, return NULL
                    }

Class Broker
{
          public static void main (string args[] )
          {
                    Time start = new Time ( );
                    VM   vm1 = new VM (RAM, MIPS, VMM, VM_AllocationPolicy, VM_SelectionPolicy);
                    Thead vm1 = new Thread (vm1);                        // Each Thread represents respective VM
                    vm1.start ( );
                    Time end = new Time ( );
                    Service time = end.getTime ( ) – start.getTime ( );      // Cloud Broker gets the service time for accounting

                    Time start = new Time ( );
                    VM   vm2 = new VM (RAM, MIPS, VMM, VM_AllocationPolicy, VM_SelectionPolicy);
                    Thead vm2 = new Thread (vm2);
                    Vm2.start ( );
                    Time end = new Time ( );
                    Service time = end.getTime ( ) – start.getTime ( );
                    .
                    .
                    .
                    .
                    .
                    Time start = new Time ( );
                    VM   vmn = new VM (RAM, MIPS, VMM, VM_AllocationPolicy, VM_SelectionPolicy);
                    Thead vmn = new Thread (vmn);
                    vmn.start ( );
                    Time end = new Time ( );
                    Service time = end.getTime ( ) – start.getTime ( );
          }
}
```

Figure 4: Implementation of Brokering Services

In our implementation step, we have considered that each Virtual Machine represents each respective user. While creating new Virtual Machine, it takes multiple parameters like RAM, MIPS, VMM, VM_AllocationPolicy, VM_SelectionPolicy as arguments. We have considered VM_Allocation Policy [23][24] such as IQR (Inter Quartile Range), LR (Local Regression), LRR (Local Regression Robust), MAD (Median Absolute Deviation), THR (Static Threshold) and VM_Selection Policy [23][24] like MC (Maximum Correlation), MMT (Minimum Migration Time), MU (Minimum Utilization). When an object of Virtual Machine has been created, we have considered starting time and finishing time of the respective thread so that cloud broker can easily keep the track of each VM and as well as each user. Cloud broker, using the data, can easily get the service time of each user and accordingly it can provide pricing model and billing services to each user. Thus, when multiple numbers of users make the requests for getting service, the respective Threads will be created according to each respective user. And cloud broker also holds the Virtual Machine ID through threads so that it keeps the accounting of each user. This is how we have considered the implementation procedure using Object Oriented Concept.

5. CONCLUSION

In the recent days, cloud computing is buzz word in academia and also in research. This has become the emerging field in the industry level. In the perspective of cloud computing, cloud broker is used computer resources to imitate another physical and computing resource and allocate those according to the requirement of the user based on SLA policies. In this paper, we have discussed the designing aspect of the cloud broker, the workflow and the sequence diagram of our design, and also implementation procedure using Object Oriented Concept. This Paper is solely based on the working technique of cloud broker and its workflow strategies. At the end of our work, we can conclude that our cloud broker Design, sequence diagram, and implementation procedure may help to develop in cloud infrastructure in the flow of rapid-growing usage of internet among the people as well as the academicians and also the future researchers.

REFRENCES:

[1] V. Sarathy et al, “Next generation cloud computing architecture -enabling real-time dynamism for shared distributed physical infrastructure”, *19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE’10)*, Larissa, Greece, 28-30 June 2010, pp. 48~53.

[2] Souvik Pal, P.K.Pattnaik, “Efficient architectural Framework of Cloud Computing”, in “*International Journal of Cloud Computing and Services Science (IJ-CLOSER)”*, Vol.1, No.2, June 2012, pp. 66~73.

[3] Farhan Bashir Shaikh, Sajjad Haider “Security threats in cloud computing”, *6th International IEEE conference on Internet Technology and secured transaction*, IEEE, 11-14 December 2012, pp. 214~219.

[4] Weiwen Zhang, Yonggang Wen and Dapeng Oliver Wu, “Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel”, IEEE Transactions on Mobile Communicatuons, Vol. 14, No. 1, January 2015, pp 81~93.

[5] Stella Gatziu Grivas, Tripathi Uttam Kumar, Holger Wache, "Cloud Broker: Bringing Intelligence into the Cloud," cloud, *2010 IEEE 3rd International Conference on Cloud Computing*, IEEE, 2010, pp.544~545.

[6] Nair, S.K.; Porwal, S.; Dimitrakos, T.; Ferrer, A.J.; Tordsson, J.; Sharif, T.; Sheridan, C.; Rajarajan, M.; Khan, A.U., "Towards Secure Cloud Bursting, Brokerage and Aggregation," *Web Services (ECOWS), 2010 IEEE 8th European Conference on* , vol., no., pp.189,196, 1-3 Dec. 2010

[7] Haresh, M.V.; Kalady, S.; Govindan, V.K., "Agent based dynamic resource allocation on federated clouds," *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE* , vol., no., pp.111,114, 22-24 Sept. 2011

[8] Hyewon Song; Chang Seok Bae; Jeun Woo Lee; Chan-Hyun Youn, "Utility adaptive service brokering mechanism for personal cloud service," *Military Communications Conference”, MILCOM 2011* , 7-10 Nov. 2011, pp.1622~1627.

[9] Smitha Sundareswaran, Anna Squicciarini, Dan Lin, “A Brokerage-Based Approach for Cloud Service Selection”, *2012 IEEE Fifth*

International Conference on Cloud Computing, IEEE Computer Society, 2012, pp. 558~565.

[10] Nikolay Grozev and Rajkumar Buyya, “Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey”, *SOFTWARE—PRACTICE AND EXPERIENCE Softw. Pract. Exper.* John Wiley & Sons, Ltd., 2012, Vol. 44, pp. 369~390.

[11] AJ Ferrer, F. Hernndez, J. Tordsson, E. Elmroth, A. Ali-Eldin,C. Zsigri, R. Sirvent, J. Guitart, RM. Badia, K. Djemame, et al., “OPTIMIS: A holistic approach to cloud service provisioning”, *Future Generation Computer Systems 2012*, Vol. 28, No. 1, pp. 66–77.

[12] Xin Jin; Yu-Kwong Kwok; Yong Yan, "Competitive Cloud Resource Procurements via Cloud Brokerage," *2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2-5 Dec. 2013, Vol. 2, pp. 355-362.

[13] Ayad Barsoum and Anwar Hasan, “Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems”, IEEE Transaction on Parallel and Distribution Systems, Vol. 24, No. 12, December, 2013, pp. 2375-2385.

[14] Simon Caton, Christian Haas, Kyle Chard, Kris Bubendorfer and Omer F. Rana, “A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks”, IEEE Transaction on Service Computing, Vol. 7, No. 3, July/September, 2014, pp. 359-372.

[15] Pal, S; and Pattnaik, P. K. (2015), “Adaptation of Johnson Sequencing Algorithm for Job Scheduling to Minimize the Average Waiting Time in Cloud Computing Environment”, *Journal of Engineering Science and Technology*. (Article in Press).

[16] Jose M. Alcaraz Calero, and Juan Gutierrez Aguado, “MonPaaS: An Adaptive Monitoring Platform as a Service for Cloud Computing Infrastructures and Services”, IEEE Transactions Service Computing, Vol. 8, No. 1, January/February, 2015, pp. 65~78.

[17] Victor Ion Munteanu, Cristina Mindruta, Teodor-Florin Fortis, “Service brokering in cloud governance”, *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, 2012, pp. 497~504.

[18] Ryan Jansen and Paul R. Brenner, “Energy Efficient Virtual Machine Allocation in the Cloud”, in *Proceedings of 2011 International Green Computing Conference and Workshops (IGCC),* 2011, pp.1~8.

[19] Souvik pal, Suneeta Mohanty, Dr. P.K. Pattnaik, and Dr. G.B. Mund, “A Virtualization Model for Cloud Computing”, in the *Proceedings of International Conference on Advances in Computer Science*, 2012, pp. 10~16.

[20] Ching-Chi Lin, Pangfeng Liu, and Jan-Jan Wu , “Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems” in the *Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing,* 2011, pp. 81~88.

[21] I. Raicu, Y. Zhao, I. Foster, and A. Szalay, “Accelerating large-scale data exploration through data diffusion”, in the *Proceedings of the 2008 International Workshop on Data-Aware Distributed Computing*, ACM, 2008, pp. 9-18.

[22] Souvik Pal, Prasant Kumar Pattnaik, “Classification of Virtualization Environment for Cloud Computing”, *Indian Journal of Science and Technology (IJST)”*, Vol. 6, Issue 1, January 2013, pp. 3965~3971.

[23] Anton Beloglazov, and Rajkumar Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", *Concurrency and Computation: Practice and Experience, ISSN: 1532-0626*, Wiley Press, New York, USA, 2011, DOI: 10.1002/cpe.1867.

[24] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, And Rajkumar Buyya “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and evaluation of Resource Provisioning Algorithms” *Software: Practice and Experience (SPE)*, Vol. 41, No. 1, January, 2011, pp. 23~50.