Resource Provisioning Using Batch Mode Heuristic Priority with Round Robin Scheduling

Gaurav Raj[1], Navreet Singh[2], Dr. Dheerendra Singh[3]
[1]PhD Scholar, Computer Science Department, Punjab Technical University
Punjab, India, [2] M. Tech., Computer Science Department, Lovely Professional University
Punjab, India, [3]Professor and Head of CSE Department , SUSCET, Tangori, India
er.gaurav.raj@gmail.com, navreet_lehal@hotmail.com, hodcse@suscolleges.com

***Abstract*—Million numbers of these users tries to access the data in different type of applications like online shopping etc., which tends to increase the load on a single server. Increase of load on a server results in reduction of throughput and it leads to a strong need of developing and maintaining an efficient system with an appropriate load balancing algorithm that will be used to retrieve the important information with a reasonable response time. The main objective of our study is to propose a load balancing algorithm that can balance the requests coming from different users residing in different locations to retrieve the data from a distributed database environment using some virtualization techniques.**

Keywords— Cloud computing, load balancing, virtualization, distributed database.

I. INTRODUCTION

Cloud computing is a style of computing, in which dynamically scalable (and mostly virtualized) resources are provided as a service over the Internet. The actual Cloud Computing definition by the National Institute of Standards and Technology is: "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Many users and consumers feel compelled to migrate to the Cloud in order to stay competitive and to benefit from the assumed improvements offered by the according infrastructure. The Cloud market is currently a high dynamic business field with new providers and business models arising effectively, therefore creating a lot of questions about what "A Cloud" actually is, and implicitly what to expect from it in the short- and long-term future. This makes this a vital time for deciding the future of Cloud computing.

Load Balancing basically ensures that all the processors in the system or every node in the network does approximately the equal amount of work at any instant of time. The load can be CPU load, memory capacity, delay or network load. Hence Load Balancing is the procedure of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work.

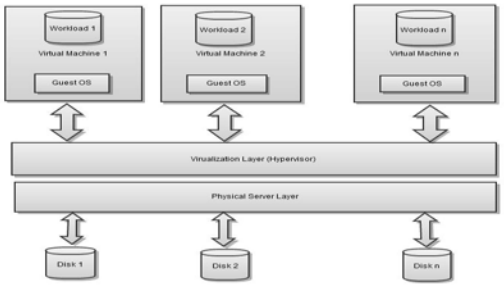

Fig.1. Layered Virtualization Technology Architecture

Today network bandwidth, less response time, minimum delay in the data transfer cost has become the main challenging issues in cloud computing. Now these challenges are also faced in the storage cloud side where in we have to balance the load across the various nodes or at the server sides. The random arrival of the load in such a cloud computing environment can cause some server to be heavily loaded. We have to maintain the load across the various servers in such a manner so that the overall performance of the servers and the network will be up to the mark and maximum at the peak level. Equally load distributing improves the performance of the

overall network by transferring the load from a heavily loaded server. Load Balancing is one of the main prerequisites to utilize full resources of parallel and distributed systems.

To optimize the performance of the cloud architecture various load balancing mechanisms should be followed in a well manner. Overloaded nodes across the server and storage side often lead to performance degradation and are more vulnerable to various failures. Therefore, in cloud computing load balancing is mainly required to distribute the dynamic load evenly and equally across all the different nodes. It helps to achieve a high user satisfaction and better resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Our main issue is to look for the proper load balancing aids that will be required in minimizing the resource consumption, starvation & over resource provisioning etc.

II. BACKGROUND: CLOUD COMPUTING APPLICATION & MECHANISMS

Cloud computing refers to Internet based development and utilization of computer technology, and hence, cloud computing can be described as a model of Internet-based computing. The cloud principles arose from a direct industrial need to improve the resource utilization without impacting on consumer requirements, i.e. use the available resources more efficiently. Initially data centers and server farms employed load management mechanisms not unlike the base Cloud principles, to ensure high availability according to the current usage. Cloud concepts promise a cost-effective realization and best methods of the utility computing principle, allowing multiple users and provider's on-demand and easy access to the various resources in a self service, pay-as-you-go fashion, thus decreasing the cost for system administration and improving resource utilization.

A. *Load Balancing*

Load Balancing basically ensures that all the processors in the system or every node in the network does approximately the equal amount of work at any instant of time. The load can be CPU load, memory capacity, delay or network load. Hence Load Balancing is the procedure of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Cloud Computing has become cost effective model for resource provisioning services and it makes the IT management easier and more responsive to the changing needs of the business. Today network bandwidth, less response time, minimum delay in the data transfer cost has become the main challenging issues in cloud computing. Now these challenges are also faced in the storage cloud side where in we have to balance the load across the various nodes or at the server sides.

Equally load distributing improves the performance of the overall network by transferring the load from a heavily loaded server. Efficient scheduling and resource allocation is a critical characteristic of the cloud computing based on which the performance of the system is estimated. All these various characteristics have a proper impact on the cost optimization, which can be obtained by improved response time and processing time. Load Balancing is one of the main prerequisites to utilize full resources of parallel and distributed systems. To optimize the performance of the cloud architecture various load balancing mechanisms should be followed in a well manner. Overloaded nodes across the server and storage side often lead to performance degradation and are more vulnerable to various failures. To remove this limitation the load must be migrated from the overloaded resource to an underutilized one without causing harm and disruption to the application workload.

Load balancing mechanisms can be broadly categorized as:

1. Centralized or Decentralized.
2. Dynamic or Static
3. Periodic or Non-periodic

B. *Load Balancing Parameters in Clouds*

The factors that always be considered in various load balancing techniques in cloud computing are as follows Detailed description of the load balancing factor is as follows:

- *Response Time* - It is the amount of time taken to provide the response by some load balancing algorithm in a distributed environment. This parameter should be minimized. It is represented as R (t).

  Formula to calculate the Response Time is:

  R (t) = Finish Time - Start Time.

  = T (f) – T(s) (1)

  Where T(f) is finish time and T(s) is start time.

- *Communication Time* - It is defined as time taken by number of hops to travel in the communication channel. It is represented by C (t). Formula to calculate the Communication Time is:

$$C(t)=2(\text{Number of hops}*\text{Time to traverse between hops}) \quad (2)$$

- *Processing Time* - It is defined as the difference between Communication Time and Response Time. It is represented by P (t). Formula to calculate the Processing Time is:

$$P(t) = \text{Response Time} - \text{Communication Time}$$
$$= R(t) - C(t) \quad (3)$$

- *Throughput* - is used to calculate the number of tasks whose execution has been completed. It should be high to improve the reliability and the performance of the system. It is represented as Th (V_i).

$$Th(V_i) = (\text{Cloudlet length}*\text{Number of cloudlets})/ \text{Response Time}$$
$$= [\text{Length}(Ci) - N_i] / R(t) \quad (4)$$

where Length (C_i) is cloudlet length and N_i is number of cloudlets for specific virtual machine.

- *Network Delay* - Delay in sending request and receiving response. It is the time taken by the network to send the number of cloudlets to particular VM and time taken by the VM to receive the cloudlets.

$$D(t) = \text{No. of cloudlets} / \text{Rate of transmission}$$
$$= N/r \quad (5)$$

where "r" is the rate of transmission.

C. *Resource Provisioning*

Resource provisioning mechanism is able to converge to an optimal or near-optimal CPU allocation within a reasonable amount of time. Secondly, it is capable of adapting even more extreme cases of resource over or under utilization. Finally, the algorithm is able to adapt to dynamically varying data rates and converges at new resource allocations in a short frame. Virtual CPU allocation should be adjusted at runtime based on the variation in workloads to achieve the goal of application **QoS** metrics.

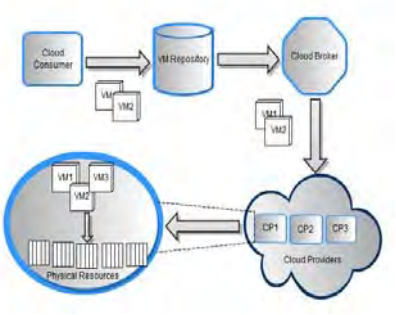

Fig. 2. System Model of Cloud Computing Environment

Resource Provisioning in the storage clouds often requires an estimate of the capacity needs of the Virtual Machines (VM's). The estimated VM size is the basis for allocating resources according to the required demands. With proper resource provisioning unused resources of a low utilized VM can be borrowed by the other co-located VM's with high utilization. Hence in cloud computing, a resource provisioning mechanism is required to supply cloud consumers a set of computing resources, for processing the jobs and storing the data.

D. *VM Life-Cycle*

The life cycle of a VM consists of the following stages:

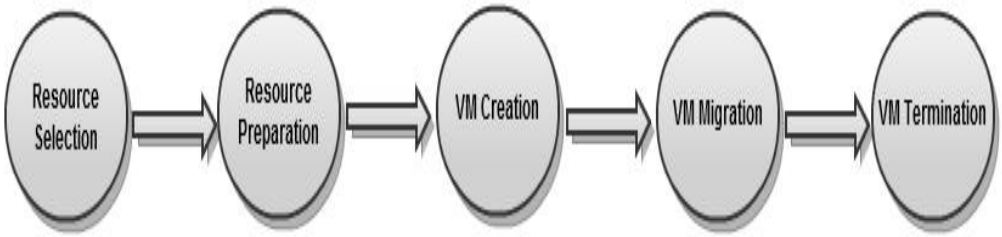

Fig. 3. Virtual Machine Life-Cycle

Resource Selection: Once a VM is requested, an efficient placement plan for the VM must be made. A scheduler provides an implementation of rank scheduling policy.

*Resource Preparation***:** The disk images of the VM are transferred to the target physical resource.

VM Creation: After the creation of every virtual machine, it is the responsibility of the virtual machine manager to allocate the resources or available cloudlets to those VMs which are recently created.

VM Migration: VMs are migrated from one location to another location in order to balance the load between the various VMs and for better resource provisioning.

VM Termination: After scheduling the cloudlet, a VM is terminated or shut down by the resource hypervisor or broker. When the VM is going to shut down, its disk images are transferred to some known location.

E. *Problem Formulation*

To optimize the performance of the cloud architecture various load balancing mechanisms should be followed in a well manner. Overloaded nodes across the server and storage side often lead to performance degradation and are more vulnerable to various failures. To remove this limitation the load must be migrated from the overloaded resource to an underutilized one without causing harm and disruption to the application workload.

In order to optimize the performance the problem is how to efficiently perform the load balancing mechanisms on the storage cloud so that the resources are better provisioned according to the user's requests. We have to use some sort of scheduling algorithm which in turn handle or divide the workload between the various nodes so that no machine will be overloaded and all the other nodes or machines are not assigned any task and are totally free.

III. THEORETICAL IMPLEMENTATION

A data center is composed by a set of hosts, which are responsible for managing VMs during their life cycles. Host is a component that represents a physical computing node in a Cloud. It is assigned a pre-configured processing capability (expressed in MIPS), memory, storage and scheduling policy for allocating processing cores to the Virtual Machines. The host component implements interfaces that support modeling and simulation of both single-core and multi-core nodes. Allocation of application specific VMs to hosts in a cloud-based data center is the responsibility of the Virtual Machine Provisioner component. There are different policies of VM provisioning based on the optimization goals. The default policy implemented by the VM provisioner is a straightforward and simple policy that allocates a VM to the host in First-Come-First-Serve (FCFS) basis.

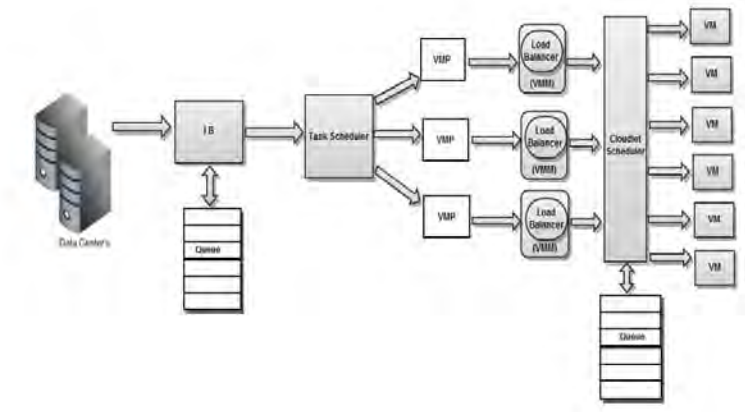

Fig. 4. Load Balancing Implementation using Task Scheduler and Cloudlet Scheduler

For each Host component, the allocation of processing cores to VMs is done based on a Host allocation. The scheduling policy takes into account that how many processing cores will be provided to each VM and how much of the processing core's capacity will effectively be attributed for a given VM. Each host component instantiates a VM scheduler component that implements the space-shared or time-shared policies for allocating cores to VMs. Cloud system developers can extend the VM scheduler component for experimenting with more custom allocation policies. But we have to choose the best scheduling and allocation policy among them.

CloudSim supports VM scheduling at two levels:

- First, at the Host level
- Second, at the VM level

At the host level, it is possible to specify how much of the overall processing power of each core in a host will be assigned to each VM. At the VM level, the VMs assign specific amount of the available processing power to the individual task units that are hosted within its execution engine.

It is required to evenly distribute the dynamic local workload across all the different nodes to achieve high user satisfaction and resource utilization. With proper load balancing, resource consumption can be kept to minimum which will further reduce energy consumption.

IV. PROPOSED LOAD BALANCING ALGORITHM

Here to solve our problem i.e. to balance the load across the various servers and for better resource provisioning we will implement a load balancing algorithm that will be a combination of two algorithms. These are:

A. Round Robin Scheduling Algorithm.

B. Batch Mode Heuristic Priority Algorithm.

These above 2 algorithms will help us to transfer the load from one sever to another in round-robin fashion and then will schedule the jobs between them so that all servers would participate in order to process the jobs and provide the better resources to execute those tasks. Now we will describe these 2 algorithm's briefly:

A. *Round Robin Scheduling Algorithm:*

Round-Robin is a simple scheduling algorithm, based on time sharing among jobs in equal slice / quantum and in circular queue without priority so it is simple and easy to implement, but it. So it focuses on fairness between jobs. The advantage of this algorithm is that no job has to wait for another one to be completed as on FCFS basis and others. However, this algorithm is not a good choice for jobs that are largely varies in their size and requirements, by means, a job is never been satisfied which in turn leads to starvation or indefinite blocking. Round Robin scheduling is usually implemented a FIFO (First in First Out) queue basis. The first element is removed from the head of the queue, scheduled, and then re-inserted at the tail of the queue. Every element is treated equally; no priorities are assigned first of all.

In this the datacenter controller assigns the requests to a list of VM'S on a rotating basis. The first request is allocated to VM which is chosen randomly from the pool of Virtual Machines and then the datacenter controller assigns the subsequent request in a circular fashion. Once the VM is assigned the request, the VM is moved to the end of the list. In this we can chose the allocation scheme that can be based on some parameters like either we can assign a weight to each of the machine or we can assign some priority to each of the VM. Based on these parameters the requests would be allocated to VM's and hence then processed. The concept of weighted Round Robin has already implemented before. So we have to choose some other parameters.

Hence In Round Robin Algorithm for load balancing, the processes are divided between all the processors. Each process is assigned to the processor in a round robin order. The order of the process allocation is maintained locally and is then independent of the allocations from the remote processors. Though the workload distributions between the various set of processors are equal but the processing time taken by each process is relatively different and they not supposed to be same at any instant of time. So, at some instant of time some of the processing nodes can be heavily loaded and other remains idle. We can also say that some resources are then over utilized and other remains underutilized which is the main drawback.

B. *Batch Mode Heuristic Priority Algorithm:*

BMHA is one of the major types of the priority scheduling algorithm. Several job scheduling algorithms have been proposed in the past years in distributed computing area. Most of them can be applied in the cloud environment. The main goal of the job scheduling algorithms is to achieve better system throughput and high

performance computing. According to the simple classification, priority job scheduling algorithms are categorized into two main groups:

- Batch-mode heuristic scheduling algorithms
- Online-mode heuristic scheduling algorithms

In BMHA, jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time and according to some priority factor. Since the cloud environment is not fixed environment and is a heterogeneous system because the speed of each processor varies quickly.

In OMHA, jobs are scheduled when they arrive in the system. The scheduling is done dynamically but in this case the waiting time to dynamically schedule the cloudlets across the VMs is more.

C. Algorithm

1. Suppose that A= {J1, J2…, Jm} is a set of jobs that request resources in a cloud environment. Also let us assume that B= {R1, R2…, Rn} is a set of resources available in cloud environment.

2. The jobs are allocated to the available resources in a round-robin fashion and also based on some priority factor.

3. We will calculate the load factor for each of the VM and corresponding to that load factor our jobs will be executed among the various VMs and resources will be allocated according to that. The resource (VM) whose load factor is more will process the job first and all the other remaining resources will remain in the idle state.

4. Batch Mode Heuristic Priority scheduling algorithm will schedule the jobs among the various VMs according to the value of load balancing factor in decreasing order.

5. Load balancing factor for each of the VM will be updated after executing each task by corresponding server which is holding the priority for executing respective task for a single instant of time.

6. We can check the status of each of the VM and can change the status of the particular VM from on to off if it is not ready to execute the jobs or if it is not willing to participate in the load balancing and resource provisioning.

7. Request Count and Active State of each of the server is maintained along with other information. The Request Count is incremented one by one whenever a single job is executed by that particular server.

8. The active state of the server can be either true or false depending upon the state of the participation of the corresponding server.

9. Jobs will be allocated to these servers according to round robin basis.

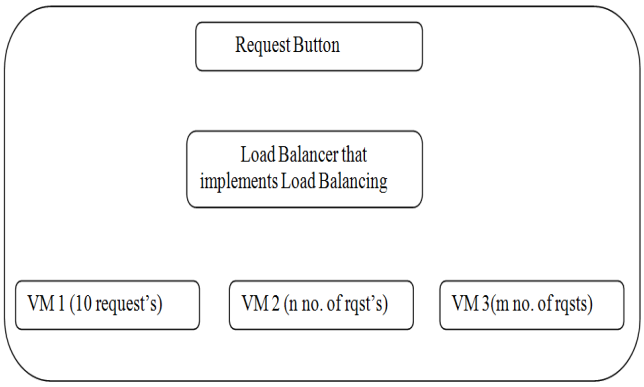

Fig. 5. Work Approach in CloudSim

However, in a virtual environment with dynamic simulation, the state of each element is constantly changing. If the element cannot get access to resource, it will accumulate an error. The main goal of our algorithm is to allot the resources (VM's) to the set of jobs according to some priority and in a round robin fashion such that workload across the storage cloud should be properly managed and hence resource provisioning would be better in cloud environment. The throughput of the whole system should be excellent and also more power should be saved for the future purpose which is the main objective of our existing approach.

D. Mechanism in CloudSim

Our mechanism in CloudSim will work as follows:

1. User can send the request by hitting the request button.

2. Load balancer decides which server is going to receive this request based on some priority criteria that is yet to be decided based on the implementing algorithm.

3. With the help of CloudSim we will create the virtual servers and their priorities. 4. User can dynamically close any server by clicking the server button and load balancer takes care of this with the help of the efficient load balancing algorithm.

5. Number of requests receive by a server are display in front server name. We can manually add and delete the number of servers.

6. Request Count is the parameter which would be maintained by each of the server. This request count would be automatically updated by the server after processing the single task.

E. Goals

1. In order to optimize the performance the problem is how to efficiently perform the load balancing mechanisms on the storage cloud so that the resources are better provisioned according to the user's requests.

2. Our algorithm can efficiently converge to the optimal CPU allocation based on the data arrival rate, priority need and computational needs.

3. Requests initiated by the users for virtual machines are submitted to the organization's cluster, but additional virtual machines are instantiated in the remote provider and added to the local cluster when there are insufficient resources to serve the user's requests.

4. The main goal of our algorithm is to balance the load across the various virtual servers and provide optimal or near-optimal CPU allocation within a reasonable amount of time.

5. Secondly, it is capable of adapting even more extreme cases of resource over or under utilization.

6. The main goal of our algorithm is to allot the resources (VM's) to the set of jobs according to some priority and in a round robin fashion such that workload across the storage cloud should be properly managed and hence resource provisioning would be better in cloud environment

V. Conclusion And Future Scope

In this paper, we have presented a modified approach of load balancing mechanism which is a combination of 2 types of scheduling i.e. round robin scheduling and Batch Mode Heuristic priority Algorithm. Our technique is used to enhance the common round robin scheduling and we have boosted up this technique by adding priority scheduling mechanism along with it whose priority is decided the based upon the load balancing factor of VM. In our future implementation, we will use Round-Robin Scheduling along with the Batch-Mode Heuristic Priority Scheduling algorithm (P_BRR). When we will use this algorithm by considering *load balancing factor* as a priority factor, the response time and throughput increases and power consumption varies based on CPU utilization. Load balancing factor is a parameter which shows that how many request a single VM can handle. We will also make the comparison of our load balancing approach with other scheduling algorithms such as Round Robin Scheduling algorithm, Priority algorithm and Online Mode Heuristic Priority with Round Robin Scheduling algorithm.

We also have to determine that on what other parameters we can further increase the performance and throughput of the system in the Cloud Computing environment as compared to our present approach. So we have to add those parameters in our existing approach by keeping in mind that this will not lead to the performance degradation of our existing approach.

References

[1] Aameek Singh, M. K, "Server Storage Virtualization: Integration Load Balancing in Data Centers," *ISSN*, 80-92 (2010).
[2] Borja Sotomayor, R. S, "An Open Source solution for Virtual Infrastructure Management in Private and Public Clouds," *IEEE*, 1-11 (2009).
[3] Jasmin James, D. B, "Efficient VM Load Balancing Algorithm for Cloud Computing Environment," *IJCSE*, 1658-1663 (2012).
[4] Jeanna Matthews, T. G, "Virtual Machine Contracts for Datacenter and Cloud Computing Environments," *ACM*, 25-30 (2009).
[5] Jennings., R, "*Cloud Computing with the Windows Azure Platform,"* Indianapolis: WILEY (2009).
[6] John W. Rittinghouse, J. F, "*Cloud Computing: Implementation, Management and Securit,"* London, New York: CRC Press (2010).
[7] Marcos Dias de Assuncao, A. d, "Evaluating the cost benefit of using cloud computing to extend the capacity of clusters," *ACM*, 141-150 (2009).
[8] Meenakshi Sharma, P. S, "Efficient Load Balancing Algorithm in VM Cloud Environment," *IJCSE, ISSN*, 439-441 (2012).
[9] Miller., M, "*Cloud Computing: Web-Based Applications That Change The Way You Work and Collaborate Online,"* Indianapolis: QUe (2009).
[10] Mohammed Alhamad, T. D, "Response Time for Cloud Computing Providers" *ACM*, 603-606 (2010).

[11] Nidhi Jain Kansal, I. C, "Existing Load Balancing Techniques in Cloud Computing: A Systematic Review," *ISSN*, 87-91 (2012).
[12] Rajkumar Buyya, J. B, "*Cloud Computing: Principles and Paradigm,"* New Jersey: WILEY (2011).
[13] Rodrigo N. Calheiros, R. R, "CloudSim: A toolkit for modeling and simulation of cloud computing environment and evaluation of resource provisioning algorithms," *D0I*, 24-50 (2010).
[14] Shamsollah Ghanbari, M. O, "A Priority based Job Scheduling Algorithm in Cloud Computing," *Proceedia Engineering*, 778-785 (2012).
[15] Smita Vijayakumar, Q. Z, "Dynamic Resource Provisioning for Data Streaming Applications in a Cloud Environment," *IEEE*, 441-448 (2010).
[16] Soumya Ray, A. D, "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment," *IJCCSA*, 1-13 (2012).
[17] Subramanian S, N. K, "An Adaptive Algorithm For Dynamic Priority Based Virtual Machine Scheduling In Cloud," *IJCSI*, 397-401 (2012).
[18] Wenying Zeng, Y. Z, "Research on Cloud Storage Architecture and Key Technologies" *ACM*, 1044-1048 (2009).
[19] Yue Tan, Y. L, "Provisioning for Large Scale Cloud Computing Services," *ACM*, 407-408 (2012).
[20] Raj Gaurav, "An Efficient Broker Cloud Management System" ACAI '11 Proceedings of the International Conference on Advances in Computing and Artificial Intelligence ACM New York, NY,USA ©2011 ISBN: 978-1-4503-0635-5.
[21] Raj Gaurav, Kaur kamaljit, "Secure Cloud Communication for Effective Cost Management System Through MSBE", International Journal on Cloud Computing: Services and Architecture, June 2012, Vol. 2, No. 3, ISSN: 2231 - 5853[Online]; 2231 - 6663 [Print]