Infrastructure Oriented Hybrid Cloud Architecture

Kamalesh Karmakar
Department of Computer Science & Engineering and Information Technology
Meghnad Saha Institute of Technology, Kolkata, West Bengal, India

Priya Roy
Department of Computer Science & Engineering
Jadavpur University, Kolkata, West Bengal, India

Abstract- Today Cloud Computing has become a key IT buzzword and mega trend, not a hype. Cloud Computing is in its infancy in terms of market adoption. In this paper Cloud Computing, its adoption issues, security issues has been discussed. A Infrastructure Oriented Hybrid Cloud Architecture has been proposed. Adoption of Public Cloud Computing services into enterprise IT includes consideration of service reliability, security, data privacy, regulation compliant requirements and so on. To address those concerns this Hybrid Cloud Computing model has been proposed. The enterprise-customers may adopt this as a viable and cost-saving methodology to make the best use of Public Cloud services along with their Private Cloud, which resides in their own infrastructure. A workload distribution service has been designed for this Integrated Cloud Computing Services. It enables and keeps track on Private and Public Cloud resources to provide on demand uninterrupted services via an Internet-based application in optimal cost. VPN has been created among VMs for secure communication.

Keywords – Hybrid Cloud, Virtual Machine Deployment-Migration-Recovery, Load Balancing.

I. INTRODUCTION

CLOUD Computing [1] has emerged as a new paradigm for providing programmatic access to scalable Internet service venues. As the world of computer science changes rapidly, users requirement increase for high computational devices and large storage space. But these may be needed for short time period. Purchasing large storage devices, high performance computing devices and licensing an application is a lengthy and tedious process to users. Cloud Computing is not a new concept. It is related to Grid Computing, Cluster Computing as well as Utility Computing paradigm. It is a computing platform for sharing resources that include infrastructure, software, application and business processes as on-demand service just on pay-per-use basis, with minimal management effort and service provider interaction.

Grid Computing [2] [3] is the collection of loosely coupled, heterogeneous & geographically distributed computer resources with non-interactive workloads. A single grid can be dedicated to a particular application, though a grid is used for a variety of purposes. Cluster Computing [4] is homogeneous, tightly coupled, collocated systems, in which single system image exists and job management and scheduling has centralized control. Cluster is used for scalable & reliable High Performance Distributed Computing [5]. Utility Computing [6] [7] focuses on the business model on which, the computing services are provided based on users requirements to provide better economics as users receive computing resources from a service provider and pay for use.

Provisioning IT services to enterprises has been burdened with complexity for users and characterized by many complex IT platform and service implementations. It is not a viable approach to IT service provisioning for results-focused business customers. Cloud provides computation environment to enable resource sharing in terms of scalable infrastructure, middleware and application development platform. The system is self-healing, SLA-driven [8], multi-tenant, service-oriented, virtualized and scalable.

Characteristics of Cloud service offering include (a) off-site data centers through third-party service providers, (b) bundled and managed Cloud solution, (c) access via Internet, using standard TCP/IP protocols, with a Web browser, (d) self provisioning and self-service requesting, (e) dynamic and fine-grained scalability (f) customization.

Cloud software leverages the Cloud model by being service oriented, with a focus on statelessness, loose application coupling, modularity, and practical semantic interoperability. Services are deployed using Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) etc.

II. Problem Statement & Motivation

Virtual Machine (VM)s deployment in Cloud Service Providers infrastructure costs lots of money. Moreover users may not want to run some application and store data in those providers' infrastructure for security reason. To maximize usage of Private Cloud resources and to run minimum number of VMs in Public Cloud, Hybrid Cloud Architecture is needed. It will provide maximum private resource utilization and minimum public resource usages. It will also provide storing and processing secure data in Private Cloud, secure communication among Private and Public VMs. Users don't need to select Private/Public Cloud environment manually. A Service Level Agreement will be done at the time of VM deployment and decision will be taken automatically at runtime if there is any failure of VM(s).

The goals of this Architecture are:

1) Cost Saving VM(s) Deployment, 2) VM(s) deployment in Private/Public Cloud from a single point of contact, 3) Automating VM(s) and Cloud resource monitoring, 4) Decision making based on statistical data of available Cloud resources and 5) Fail-over recovery and VM migration without users' intervention.

III. Hybrid Cloud

Public or External Cloud describes Cloud Computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web services and web applications, from an offsite third-party provider who shares resources and bills on a fine-grained Utility Computing basis. Private or Internal Clouds are recently used to describe offerings that emulate Cloud Computing on private networks infrastructure.

The Hybrid Cloud is capable to provision, monitor, and move virtual servers. Part of its purpose is to let Private Cloud users get services from external Clouds and establish links with service providers via this Hybrid Cloud API to send a workload to an external Cloud or enable a workload in an external Cloud to tap into services that are part of the enterprise infrastructure. This Hybrid Cloud infrastructure is a composition of Private and Public Cloud that are bound together by standardized or proprietary technology that enables data and application portability with specialized access and security requirements.

The key factors of this architecture are resource-pooling (to serve multiple users using a multi-tenant shared model, with different physical and virtual resources assigned dynamically), location-independence (the customer generally has no control or knowledge over the exact location of the provided resources), resource-abstraction, elasticity (provisioned to scale out and rapidly released to scale in). This model generates and stores reports on resource-usage that is monitored, controlled providing transparency to user and service provider. To provide security and privacy VPN has been used.

The architecture of a Cloud Computing includes several key modules. The system manages massive network of servers running in parallel using virtualization techniques to dynamically allocate and de-allocate computing resources. The target is to hybridize the application to distribute workloads in Private and Public Cloud with scale-up and scale-down. The HCA model allow enterprises seamlessly and securely extend their Private Clouds to Public Cloud and higher priority applications are given preference over the lower priority ones when resources become constrained. After installation it could seamlessly roll client workloads from one geographical location to another to ensure that clients never miss a Service Level Agreement (SLA).

HCA provides full benefits of Cloud Computing using common APIs to transition between Private Cloud and Public Cloud to push just enough data into the Cloud to perform a computation and obtain an acceptable result, or seamlessly pull in resources from a Public Cloud when local capacity is temporarily exceeded. Users can use either the Private Cloud or the Public Cloud as a spare in the event that one VM becomes unavailable or fails.

IV. Cloud Environment Setup

In this Hybrid Cloud environment Eucalyptus (an open source Private Cloud building tool) has been used as Private Cloud and Amazon EC2 has been licensed for accessing Amazon Cloud. In Eucalyptus, Java and Web services create premises-based Cloud Computing infrastructure using their own hardware and software. It is nearly re-implementation of the Amazon Web Services EC2 API. Eucalyptus makes datacenter compatible at the API level like an Amazon Cloud.

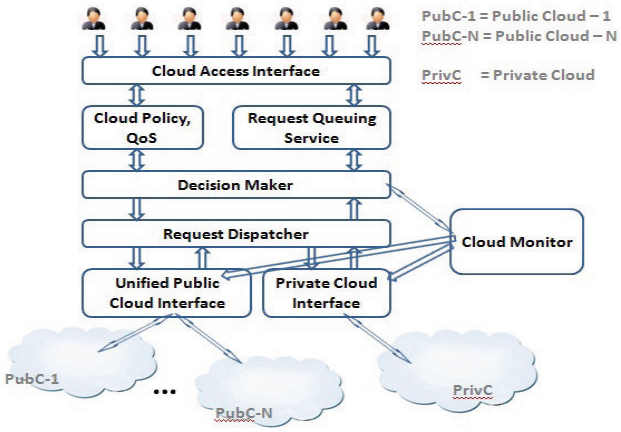

Figure. 1: Infrastructure Oriented Hybrid Cloud Architecture

In this infrastructure two Eucalyptus Clouds has been set up, configuring multi-cluster with two availability zones, to allocate resources for VMs in any Cloud environment based on available resources. Storage Controller is running on each Cluster for EBS like storage. Walrus has been installed to store persistent data, organized as buckets and objects. In the middleware Amazon EC2 API has been integrated to deploy VMs in Amazon Cloud environment.

V. Proposed Hybrid Cloud Architecture

In this research work a web based user interface has been developed to deploy VMs in Private or Public Cloud or in both environments. Users don't need to bother about manual decision making to minimize the cost, based on present Cloud status.

Http serves as a request and response procedure that all agents on the Internet follow so that information can be rapidly, easily, and accurately disseminated between servers, which hold information, and clients, who are trying to access it. Using this middleware's web interface users is exchanging confidential information with server, which needs to be secured in order to prevent unauthorized access. For this reason, this middleware's web interface has been deployed using https protocol to transfer sensitive information securely.

HCA consists of many modules shown in the Figure-1. The necessity and functionality of those modules will be discussed below.

A. Cloud Access Interface (CAI) –

It is an attempt to create an open and standard Cloud interface for communicating with different types of Cloud environments (provided by different Public Cloud service providers and Internal Private Cloud environments). This is a single programmatic point of contact that can encompass the entire infrastructure stack as well as emerging Cloud centric technologies all through a unified interface.

CAI acts as a brokering interface so that users can easily deploy their system to Cloud environment to get benefits of Hybrid Cloud. It is composed of a specification and schema. The schema provides the actual model descriptions, while the specification defines the details for integration with other management models. This is a common interface for the interaction with remote platforms, systems, networks, data, identity, applications and services. A common set of Cloud definitions enable vendors to exchange management information between remote Cloud providers using secure global asynchronous communication.

1) Registration, Authentication and Authorization: Before accessing this Hybrid Cloud two types of registrations are required, User-Registration and Cloud-Registration. In User-Registration user has to register his personal information and in Cloud-Registration user has to provide Cloud name, Cloud access information, like access-key, Cloud-IP/port schemaversion. To access this Hybrid Cloud user has to provide proper authentication parameter values for verifying the claim made by him. On the other hand, Authorization involves verifying that an authenticated person has permission to perform certain operations or access specific resources.

2) Service Level Agreements to Access Clouds: A SLA is defined in a contract or an agreement that outlines service usage and guarantees the service-level capabilities. The QoS concerns focus on measurable, performance-

oriented factors, such as availability and responsiveness. Different parameters, which are considered in accessing Clouds, are defining security level, security group, and performance factor.

B. Service Request Admission and Admission Control (SRAAC)

SRAAC provides API integration to deploy VMs requested through CAI. It maintains log files, stores VM status, stores scripts (which ran on it), stores monitoring information etc. SRAAC provides request queuing, policy checking, decision making and monitoring services. SRAAC contains Request Queuing Service, Cloud Policy, Cloud Monitor, Decision Maker and Request Dispatcher modules. The functionality of these modules are described below.

1) Request Queuing Service (RQS): It has been introduced to offer a reliable, highly scalable, hosted queue for storing messages as they travel between computers. The system can simply move data between distributed components of their applications that perform different tasks, without losing messages or requiring each component to be always available.

RQS makes it easy to build an automated workflow, working in close conjunction with the Eucalyptus, Amazon EC2 and the other AWS infrastructure web services. When a request is submitted, it is maintained in queue and decision maker retrieves the request to process. This is a priority based queue. If decision maker checks that currently available resources are not sufficient to process the request, it does not erase the job request from queue. When decision maker decides to deploy the request and transfer it to request dispatcher, it erases the request from queue.

A request is represented as follows:

Reqi = { Ji , Sk, Pi }

[Ji : i-th Job Details, Sk : Service Level Agreement for Provider k, Pi : Priority]

2) Cloud Policy and Quality of Services (QoS): Cloud Policy is Service Level Agreement (SLA) between users and service providers. Different organizations provide their services in different level and at different cost and QoS is also considered. To build a hybrid Cloud, web servers, databases are bundled up and configured that make up the application, which is known as deployment. This deployment is portable and able to run in a Public Cloud or in-house, because it defines not only the components but also how they interact and grow. Then rules are set as where and how that deployment can run according to specific security, capacity and business requirements. Depending on users SLA this module sets policies as to which applications can run where, such as a Cloud must provide in order for a particular deployment to run.

SLA is represented as follow:

SLA = {T, P, R, A, S, B}

[T : Time reservation to run VMs (for Private Cloud it is infinite), P : Priority defined for service, R : Responsibilities, A : Availability level, S : Scalability, B : Billing]

Lots of challenges come as how big or small the application can be, how to bind a component to the available resources, how to grow the deployment when certain events happen.

3) Decision Maker: It is connected to Cloud Monitor and Cloud Policy. Depending on user request, DM retrieves information about policy, SLA, pricing etc from Cloud Policy and it retrieves information about current status of Clouds from Cloud Monitor. Then these all information is processed by it.

It decides in which Cloud (Private or Public) the request will be executed depending on users expected security defined in SLA. If very much secure part is needed to be processed these are executed in Private Cloud. It may happen that at that time required resource is not free in Private Cloud, it will wait in RQS. Other applications can run in Public Cloud. Depending on sensitivity applications are categorized in two types, one is less-sensitive and other is highly-sensitive, which are deployed in Public and Private Cloud respectively. Here less-sensitive does not mean an application which is not secure. It means the user want to execute it in Public Cloud because it does not contain very sensitive data which may be visible to other people outside the Private Infrastructure as the control of Public Cloud is in provider's hand.

DM collects following Resource information from Private Cloud:

Cloud Regions {AZ, SG}

AZ : Availability Zones, SG : Security Groups

Cloud Resources {M, D, C}

MA : Available Memory, DA : Available Disk, CA : Available CPU Core

To deploy a VM, DM concentrates on these parameters:

```
If CR <CA & MR <MA & DR <DA
  deploy VM in Private Cloud.
Else
  if SG & SL !ϵ SLApublic
        wait in RQS for required available Private-Cloud resource.
  else
        deploy VM in Public Cloud
```

[where MR : Required Memory (NI * MI), DR : Required Disk, CR : Required CPU Core, ITI : Instance Type, NI : Number of Instances, MI : Required Memory for a Instance of type ITI, SL : Security Level, SG : Security Group, Cloud Region, Key-Pair]

Based on security, data is divided into three categories, those are:

1) Highly Confidential (HC), 2) Confidential (C) and 3) Less Confidential (LC)

Highly trust worthy information processing is labeled as Highly Confidential, like storing, processing and generating key information. Private-Cloud is considered as highly trust worthy information processing system. Generally processing jobs are Less Confidential which does not require processing of confidential data. These systems can run in public-Cloud when Private Cloud resources are not free. Confidential systems are those, which processes or stores confidential data but not like authentication keys. These systems can run in Private/Public Cloud depending on priority and execution deadline and waiting time exceeds threshold.

4) Request Dispatcher: Request Dispatcher invokes Unified Cloud Interface. Decision Maker prepares the information about VM deployment and sends it to Request Dispatcher.

The information comes to RD in the form of

Req dispatcher {IT, N, CLD, C, M, D, K, SL, SG}

[IT : Instance Type, N : Number of Instances, CLD : Cloud Regions, C : CPU Core, M : Memory, D : Disk, K : Key-Pair, SL : Security Level, SG : Security Group.]

RD dispatches the request to the corresponding Cloud calling the API and schema-version. It records the deployment time, service up time on VM etc. MySQL has been used as Database Server to store all these monitoring and deployment information.

5) Unified Cloud Interface: A user has access to Private Cloud infrastructure and Public Cloud such as Amazon EC2, GoGrid etc. This user normally accesses the Cloud infrastructure through multiple access programs with multiple credentials. UCI integrates different types of Cloud APIs to access multiple Cloud service providers' Cloud-platform. Typica has been modified to access Eucalyptus Cloud and Amazon EC2.

C. Cloud Monitor

Cloud Resources are defined as:

Resource {Memory, CPU Core, Disks}

CM dynamically collects information about registered Clouds (Private and Public), CPU usage, memory usage, and etc of all the Clouds and health of VMs running on Clouds. Cloud Monitor empowers to proactively monitor VM deployment, resource usages and determine actionable next steps such as proactive outreach or creation of a service case based on information gathered. CM creates and maintains log files for each different monitoring service. It stores collected monitoring information in DB server.

In HCA monitoring task is done in the following way.

1) Cloud Environment Monitoring (CEM): a) Accessibility of registered Clouds: whether it is connected to internet and accessible, b) Resource usage: how much resources are being used and how much is available to be used, c) VM status monitoring and d) Virtual Disks and Volumes Monitoring.

2) Virtual Machine Monitoring (VMM): a) Service Monitoring, b) Process Monitoring, c) Memory usage Monitoring, d) Disk Monitoring, e) CPU Load Monitoring and f) Performance Monitoring.

For collecting statistics (Profiling) about how many requests are being processed and how it is being handled, the polling interval has been set to such a value that network traffic is low. Cloud Monitor runs in 'Monitor Poll' mode. In this mode some processes in the monitoring system poll the system elements in some thread. During the poll hosts are accessed via SSH to execute scripts and OS specific commands and state-output-files are sent back to monitor.

The most important job of CM is reporting to Decision Maker and for unrecoverable failure reporting to users, generating mail to System Admin using JMS API. A monitoring tool has been designed for Cloud environments and VM(s). When a new instance gets an IP address, it is registered to Monitoring tool. Shell scripts are run on VMs using SSH to retrieve system status (like process, memory, disk, services etc.).

Log files basically contain status information which is not being reused for further processing. The information which is needed for further processing and decision making are stored in DB server.

VI. VM DEPLOYMENT, MIGRATION AND RECOVERY TECHNIQUE

VM Deployment technique has been shown in Figure.2 and VM Recovery technique has been shown in Figure.3.

User requests in CAI and agrees to Cloud Policy depending on his/her requirement. Then the information about requirement, policy & QoS goes to Decision Maker. Decision Maker then retrieves current status of Private Cloud. It then decides where to deploy. If Cloud policy restricts to run VM in Public Cloud, request is maintained in Request Queuing Service. When resources in Private Cloud become available it is deployed to the Private Cloud.

Cloud Monitor retrieves status of Private Cloud and detects failure if any. And it sends information to Decision Maker. Getting this failure information Decision Maker retrieves Policy, QoS etc from Cloud Policy and decides where to deploy the VM. If policy is not restricted very much VM can be migrated from Private to Public Cloud. If private resources are free or available then VMs can be migrated from Public to Private Cloud, because Public resources are used on payment basis.

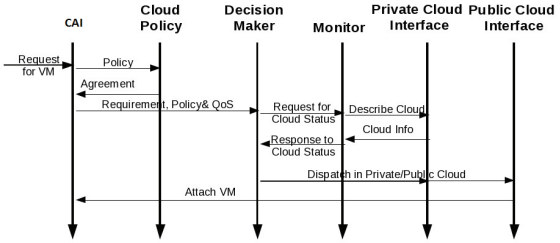

Figure. 2: VM Deployment Technique

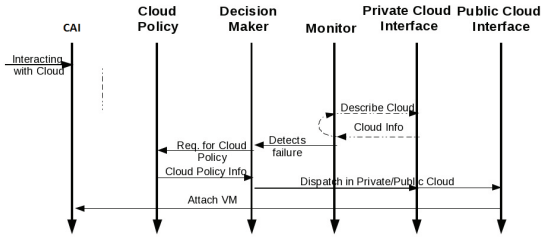

Figure. 3: VM Recovery Technique

By this Hybrid Cloud interface (shown in Figure.4) Eucalyptus Cloud can be registered and Amazon Public Cloud can be accessed. VMs can be deployed in different Cloud environments depending on available resources.

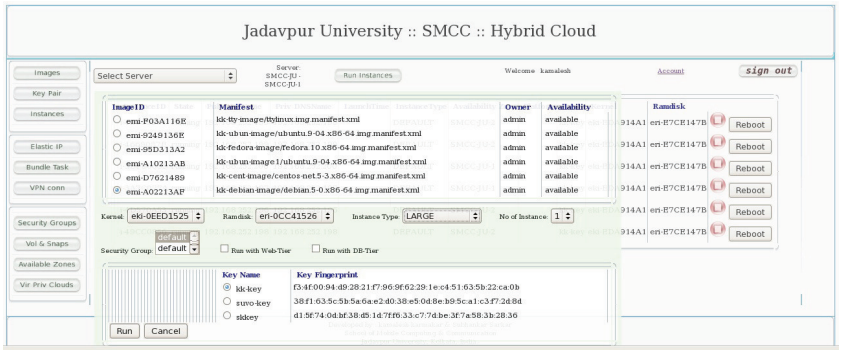

Figure. 4: Hybrid Cloud Management Interface

If one VM fails at any moment then the remaining application runs in other VM and Decision Maker takes corresponding steps to assign that part of an application to other Cloud. The new VM is booted with the same SLA and applications run on it based on the log which is stored in the database. Based on different application types, some application may be started from the check-point and some application may need restart. It sometime affects performance but guarantees fail-over recovery.

VII. LOAD BALANCING

The Cloud's scalability means if any application suddenly needs a lot more resources than anticipated, the Cloud can easily allocate (scale-up) more resources to it. Since applications aren't tied to any one discrete server, it can also deallocate (scale-down) resources which are not being used. The entire Cloud needs to scale if the Cloud overloads or goes down, users can outsource CPU cycles to a different Cloud. Load balancing is the technique to provide scalability on rapid provisioned underlying system. Load balancing is applied as the demand for Web Services fluctuates; as it is desirable to dynamically allocate VMs to regulate the system against demand.

In Cloud environment, a collection of virtual servers runs into physical servers to provide QoS at minimal cost. This load balancing technique measures how effectively resources are being used. Depending on domain-specific requirements, this may be a simple aggregation of individual VM's CPU idle time, or may encompass other metrics.

VM(s) runs independently by different virtual processors. In this model load-balancing management application runs in different system (other than Cloud environment).

There are three levels of decision making in workload distribution algorithm as follows.

1) Deciding in which Cloud it can be deployed with high priority of Private Cloud.

2) Deciding in which cluster it can be deployed.

3) How resources can be managed scaling up/down the VMs.

In this Auto Scale policy, Load Balancer collects the information from VMs and Cloud environments as Triggered Action. Load Balancer sends message to CM to be quiet for few seconds. Load Balancer reads Cloud Deployment policy and QoS parameters and initiates balancing algorithm to run to make a stable state before any action can take place.

VIII. SECURE COMMUNICATION AMONG VMS

A. Security Issues: Cloud Computing is picking up traction with businesses, but there are some unique security risks. Cloud Computing is fraught with security risks. One type of solution to data security is getting a security assessment from a neutral third party before committing to a Cloud vendor. Cloud Computing has unique attributes that require risk assessment in areas such as data integrity, recovery and privacy, and evaluation of legal issues in areas such as e-discovery, regulatory compliance and auditing. The questions come related to the policy makers, architects, coders and operators; risk-control processes and technical mechanisms; and the level of testing that's been done to verify that service and control processes are functioning as intended, and that vendors can identify unanticipated vulnerabilities.

Here are some security issues, such as Privileged user access, Regulatory compliance, Data location, Data segregation & Recovery, Investigative support, Long-term viability etc which users raise with vendors before selecting a Cloud vendor. In this work data security is considered during data transfer among geographically distributed VMs. Every user's data should be segregated from other user's data.

B. Virtual Private Network (VPN): In this architecture IPSec has been used to setup VPN. It provides a secure communication path between VMs in two Private Cloud, between Private & Public Cloud and between Public Clouds. This secure channel always involves a Request Dispatcher as enabler of secure communication between machines and the trusted authorities of different Clouds. Security in this context means providing authentication of the requester and confidentiality, integrity, and data-authentication services for the data sent across the channel. This secure channel enables secure replication of the machine images and secures exchange of Challenge/Response messages and pass-through authentication.

VPN is a computer network in which some of the links between nodes are carried by open connections or virtual circuits in some larger networks, as opposed to running across a single private network. The Link Layer protocols of the virtual network are tunneled through the transport network. One common application is to secure communications through the public Internet, but a VPN does not need to have explicit security features such as authentication or content encryption. VPNs can also be used to separate the traffic of different user communities over an underlying network with strong security features.

In this architecture IPSec has been used to protect every application-traffic in IP Network. IPSec has been chosen as it is very established protocol and is well supported by pretty much anything that supports VPN connections (router, Operating Systems, smart phones etc.). IPSec makes it reasonably easy to secure what can and cannot go over a tunnel, at the kernel level, without having to set up extra firewall rules.

IPSec protocol suite secures communication by authenticating and encrypting each IP packet of a communication session. In this set up IPSec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of encrypted keys to be used during session. IPSec 'Tunnel Mode' has been used so that entire packet is encrypted and authenticated and data flows only between a pair of hosts, between a pair of security gateways or between security gateway and host.

IX. CONCLUSION

Cloud Computing is an emerging computing paradigm that is increasingly popular. Leaders in the industry, such as Amazon, Rackspace, Google, and IBM, Microsoft etc have provided their initiatives in promoting Cloud Computing. Instead of building applications on fixed and rigid infrastructures, HCA provides a new way to build applications on on-demand infrastructures without having any upfront investment to run a job massively distributed on multiple nodes in parallel and scale incrementally based on the demand with full transparency.

The modules of Hybrid Cloud Model have been designed to overcome critical situations. If one VM fails, another VM of the same image is launched with previously running services within very short time. In this way uninterrupted service can be provided to user. When Private Clouds are overloaded VM deployment is done over Public Clouds. All user requests are maintained in queue and deployed accordingly. Security is a major problem in Cloud Computing environment as systems are fully distributed in different geographical locations. For secure communication IPSec has been used. Auto deployment and recovery is also a big problem in Cloud environment. Users activity log is maintained to keep track on user activities. When some system fails this is taken care that user is not be interrupted, so depending on SLA, systems are deployed automatically.

X. FUTURE WORK

In this research Eucalyptus and Amazon EC2 APIs have been integrated. It has been planned to integrate other Cloud Computing environments like OpenNebula, Rackspace etc. Different CSPs have different Service Level Agreements, APIs, tools and different specification, for this reason for more transparency to users we need to integrate more Cloud environments. We want to work also on other security measures in future as security is a big problem in Cloud environment. In future we are planning to work on unplanned outages and failures of VM.

REFERENCES

[1] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility.

[2] Daniel A. Menasce, Emiliano Casalicchio, A Framework for Resource Allocation in Grid Computing (MASCOTS'04) .

[3] Zhimin Tian, Yang Yang, Zhengli Zhai, Modelling Robust Resource Allocation for Grid Computing. Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), 0-7695-2694-2/06 Âl' 2006.

[4] Mino Ku; Dugki Min; Eunmi Choi, SCAREX: A framework for scalable, reliable, and extendable cluster computing, Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference.

[5] Manish Parashar, Salim Hariri, A. Gaber Mohamed and Geoffrey C. Fox. A Requirement Analysis for High Performance Distributed Computing over LAN. 0-8186-2970-3192 1992 IEEE.

[6] Pradeep Padala, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, Kenneth Salem, Adaptive Control of Virtualized Resources in Utility Computing Environments, EuroSysâ ˘A ´ Z07, March 21â˘A ¸S23, 2007, Lisboa, Portugal. ACM 978-1-59593-636-3/07/0003

[7] Hangwei Qian, Miller, E. ; Wei Zhang, Rabinovich, M., Wills. Agility in Virtualized Utility Computing, Virtualization Technology in Distributed Computing (VTDC), 2007 Second International Workshop.

[8] MinChao Wang, Xing Wu ; Wu Zhang ; FuQiang Ding ; Jun Zhou ; GuoCai Pei. A Conceptual Platform of SLA in Cloud Computing, Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference.