A Project report
on

SEAMLESS PROVISION OF CLOUD SERVICES USING PEER-TO-PEER (P2P) ARCHITECTURE

Submitted to

Department of Computer and Information Sciences,

In partial fulfillment of the Degree of Master of Technology in Information Technology

Submitted by

Snehal Masne

Under the guidance of

Dr. Rajeev Wankar
(Project Supervisor)

Prof. C R Rao
(Project Supervisor)

Department of Computer and Information Sciences,
School Of Mathematics Computer & Information Sciences
University Of Hyderabad

April, 2011

CERTIFICATE

This is to certify that the project work entitled - 'Seamless Provision Of Cloud Services Using Peer-To-Peer (P2P) Architecture' accomplished by Snehal A. Masne, bearing Reg. No. 09MCMB20 in partial fulfillment of the requirements for the award of Master of Technology (Information Technology) by the University of Hyderabad, has carried out the work under my supervision and has submitted a satisfactory account of report in this thesis.

The results embodied in this thesis have not been submitted to any other University or Institution for the award of any degree or diploma.

Dr. Rajeev Wankar
Project Supervisor,
Department of CIS,
University of Hyderabad

Prof. C R Rao
Project Supervisor,
Department of CIS,
University of Hyderabad

Head,
Department of CIS,
University of Hyderabad

Dean,
School of MCIS,
University of Hyderabad

DECLARATION

I, Snehal Masne, hereby declare that this Dissertation entitled "Seamless Provision of Cloud Services Using Peer-To-Peer (P2P) Architecture" submitted by me under the guidance and supervision of Dr. Rajeev Wankar and Prof. C R Rao is a bonafide work.

I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date : 25/04/2011

Signature :

Snehal Masne
M.Tech IT
Reg. No. 09MCMB20

ACKNOWLEDGEMENT

I had been working for this day. The completion of this project work has given me satisfaction and pleasure I had been working for.

It is a privilege to express my profound gratitude and indebtedness to my supervisors Dr. Rajeev Wankar and Prof. C R Rao for their valuable, inspiring and constant support throughout the progress of this project. Their suggestions helped me to improve my work a lot.

I'm grateful to the Head of Department (DCIS), Prof. C R Rao for providing both the freedom and an excellent lab facility where I could do and test project work effectively.

I sincerely thank all those who have helped us in the endeavor of completing the project. I would also like to thank all my fellow colleagues who readily helped me directly or indirectly whenever I was in need.

I express my gratitude to all my family members for their love and affection, without which this work would not have been possible. And last but not least, I am thankful to the almighty. It was His blessings that helped me till the end.

Snehal Masne
M.Tech Information Technology
Department of CIS,
University of Hyderabad

ABSTRACT

In recent years clouds have unleashed the next generation computing that facilitates creation of wide-area on-demand renting of services (Infrastructure, Platform and Software). Cloud computing involves highly variable resource requirement that demands high availability, scalability and performance. At times, single cloud service provider would be saturated or running out of resources and may be unable to provide the service to its client, resulting in poor scalability and reliability. It may tarnish the trust parameter of customer. Also there is huge investment in setting up a single scalable cloud, which in turn has many environmental impacts.

In this project, we propose a novel architecture to inter-connect different clouds in P2P fashion to address the problems bottleneck and single point of failure that are predominantly associated with traditional approaches. This innovative idea allows sharing of own local resources/services with other partner cloud providers and hence can get access to much larger pools of resources/services. Also each provider can maximize their profit by creating new collaborative services. These capabilities can be available and tradable through a service catalogue to support new innovations and applications.

Keywords *- Cloud scalability, Interoperability, Cloud peer-to-peer (P2P) architecture.*

TABLE OF CONTENTS

Chapter - 1

INTRODUCTION

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

1.2. Why Cloud Computing?

Cloud computing has the potential to upend the software industry entirely, as applications are purchased, licensed and run over the network instead of a user's desktop. This shift will put data centers and their administrators at the center of the distributed network, as processing power, electricity, bandwidth and storage are all managed remotely. It affects not only business models, but the underlying architecture of how we develop, deploy, run and deliver applications.

1.3. Glimpse of our project

Cloud computing involves highly variable resource requirement that demands high availability, scalability and performance. At times, single cloud service provider would be saturated or running out of resources and may be unable to provide the services to its client, resulting in poor scalability and reliability. It may tarnish the trust parameter of customer. Also there is huge investment in setting up a single scalable cloud, which in turn has many environmental impacts.

In this work, we propose an architecture to inter-connect different clouds in P2P fashion to address the problems like efficiency bottleneck and single point of failure that are predominantly associated with traditional approaches. This idea allows sharing of own local resources/services with other partner cloud providers and hence can get access to much larger pools of resources/services. Also each provider can maximize their profit by creating new collaborative services. These capabilities can be available and tradable through a service catalogue to support new innovations and applications. This project also explores the issues related to cloud interoperability which plays profound role in simplifying the maximum and efficient utilization of resources.

1.4. Organization of Project

As mentioned earlier, our work deals with solving the problem of scalability and interoperability. This complete project is organized into following steps: exposing to cloud computing, analysis of current issues, understanding the exact problem, designing of the solution architecture, addressing the minute details of the interoperability, use of proper protocols/standards and so on. This systematic approach helped us to arrive at final and feasible solution that envisions tomorrow’s cloud computing. We finished this project in one year and gave enough time for aforementioned steps.

Chapter - 2

LITERATURE SURVEY

2. LITERATURE SURVEY

In this literature survey, we try to explore all nuts and bolts of cloud computing. Grasping the details of cloud computing aids in resolving the problem right way. In this section we get acquainted with Cloud Characteristics, Benefits and Risks, Service delivery models, Cloud Deployment models, Cloud Internals (UEC), etc.

2.1. Essential Cloud Characteristics

Here we list some of the major essential characteristics of a cloud [1]:

A. On-demand self-service

The self-service cloud is a powerful IT service delivery model that significantly reduces administrative overhead while driving end-user productivity and satisfaction. Services are delivered on demand or reserved in the future for end users in a completely self-service manner. As a part of this project we have developed on demand (based on availability of instance slot) service to access the cloud. Here the admin can assign slot based on user request. To make it more advanced cloud, the future work would be making reservations to the user, so that the cloud becomes more easy way to access.

B. Ubiquitous network access

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g. mobile phones, laptops, and PDAs).

C. Resource pooling

The provider's computing resources are pooled to serve all consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction.

D. Rapid elasticity

Capabilities can be rapidly and elastically provisioned to quickly scale up, and rapidly released to quickly scale down based on demand of requirement.

E. Measured service

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency of the utilized service for both the provider and user.

2.2. Benefits and Risks of Cloud Computing

Here are some benefits of cloud computing [1]:

- **Reduced Cost:** Cloud technology is paid incrementally, saving organizations money.
- **Increased Storage:** Organizations can store more data than on private computer systems.
- **Highly Automated:** No longer do IT personnel need to worry about keeping software up to date.
- **Flexibility:** Cloud computing offers much more flexibility than past computing methods.
- **More Mobility:** Employees can access information wherever they are, rather than having to remain at their desks.
- **Allows IT to Shift Focus:** No longer having to worry about constant server updates and other computing issues, government organizations will be free to concentrate on innovation.

Following are some risks in cloud computing

- **Privileged user access:**

  Sensitive data processed outside the enterprise brings with it an inherent level of risk, because outsourced services bypass the physical, logical and personnel controls. IT shops exert over in-house programs. So it becomes essential to get as much information as you can about the people who manage your data, ask providers to supply specific information on the hiring and oversight of privileged administrators, and the controls over their access.

- **Regulatory compliance:**

  Customers are ultimately responsible for the security and integrity of their own data, even when it is held by a service provider. Traditional service providers are subjected to external audits and security certifications. Cloud computing providers who

refuse to undergo this analysis are signaling that customers can only use them for the most trivial functions.

- **Data location:**

  When we use the cloud, we probably won't know exactly where our data is hosted. In fact, we might not even know what country it will be stored in. It's better to ask providers if they will commit to storing and processing data in specific jurisdictions, and whether they will make a contractual commitment to obey local privacy requirements on behalf of their customers.

- **Data segregation:**

  Data in the cloud is typically in a shared environment alongside data from other customers. Encryption is effective but isn't a cure-all. The cloud provider should provide evidence that encryption schemes were designed and tested by experienced specialists. Encryption accidents can make data totally unusable, and even normal encryption can complicate availability.

- **Recovery:**

  Even if you don't know where your data is, a cloud provider should tell you what will happen to your data and service in case of a disaster. Any offering that does not replicate the data and application infrastructure across multiple sites is vulnerable to a total failure. Check if it has the ability to do a complete restoration, and how long it will take.

- **Investigative support:**

  Investigating inappropriate or illegal activity may be impossible in cloud computing. Cloud services are especially difficult to investigate, because logging and data for multiple customers may be co-located and may also be spread across an ever-changing set of hosts and data centers. If you cannot get a contractual commitment to support specific forms of investigation, along with evidence that the vendor has already successfully supported such activities, then our only safe assumption is that investigation and discovery requests will be impossible.

- **Long-term viability:**

  Ideally, your cloud computing provider will never go broken or get acquired and swallowed up by a larger company. But you must be sure your data will remain available even after such an event.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

1.1.2. Platform as a Service (PaaS)

PaaS can be described as an entire virtualized platform that includes one or more servers (virtualized over the set of physical servers), operating systems, and specific applications (such as Apache and MySQL for Web-based applications). In some cases, these platforms can be predefined and selected; in others, you can provide a VM image that contains all the necessary user-specific applications. Types:

• Add-on development facilities: These facilities allow customization of existing SaaS applications, and in some ways is the equivalent of macro language customization facilities provided with packaged software applications such as Lotus Notes, or Microsoft Word. Often these require PaaS developers and their users to purchase subscriptions to the co-resident SaaS application.

• Standalone development environments: Stand-alone PaaS environments do not include technical, licensing or financial dependencies on specific SaaS applications or web services, and are intended to provide a generalized development environment.

• Application delivery-only environments: Some PaaS offerings lack development, debugging and test capabilities, and provide only hosting-level services such as security and on-demand scalability.

• Open Platform as a Service: Lets the developer use any programming language, any database, any operating system, any server, etc.

As we are not dealing with SaaS and PaaS in this project we limit the discussion till here. We will see IaaS in detail next.

1.1.3. Infrastructure as a Service (IaaS)

IaaS is the delivery of computer infrastructure as a service. This layer differs from PaaS in that the virtual hardware is provided without a software stack. Instead, we have to provide a VM image that is invoked on one or more virtualized servers. IaaS provides and maintains the underlying hardware, operating system and network Infrastructure resources and provides it in a virtualized, easy to manage commoditized way.

IaaS is the unrefined form of “Computing as a service” or also called as “Hardware as a service” as in our cloud we are providing the hardware space for instance to run and perform other operations.

Here we have built a Eucalyptus cloud using UEC, where we provide VM (operating system as of now), so that the user can access the deployed OS images in the

cloud. As a provider, we give hardware space and computing power to user. Here eucalyptus is an open source implementation of Amazon EC2 that is interface-compatible with the commercial service. Like EC2, Eucalyptus relies on Linux with Xen for operating system virtualization.

2.4. Deployment models

Organizations can choose to deploy applications on public, private, or hybrid clouds [1], each of which has its trade-offs. The terms public, private, and hybrid do not state location. While public clouds are typically "out there" on the Internet and private clouds are typically located on premises, a private cloud might be hosted at a collocation (side-by-side) facility as well.

Organizations may make a number of considerations with regard to which cloud computing model they choose to employ, and they might use more than one model to solve different problems. An application needed on a temporary basis might be best suited for deployment in a public cloud because it helps to avoid the need to purchase additional equipment to solve a temporary need. Likewise, a permanent application, or one that has specific requirements on quality of service or location of data, might best be deployed in a private or hybrid cloud.

1.2.1. Public Cloud

Public clouds are run by third parties, and applications from different customers are likely to be mixed together on the cloud's servers, storage systems, and networks. Public clouds are most often hosted away from customer premises, and they provide a way to reduce customer risk and cost by providing a flexible, even temporary extension to enterprise infrastructure. If a public cloud is implemented with performance, security, and data locality in mind, the existence of other applications running in the cloud should be transparent to both cloud architects and end users. Indeed, one of the benefits of public clouds is that they can be much larger than a company's private cloud might be, offering the ability to scale up and down on demand, and shifting infrastructure risks from the enterprise to the cloud provider, if even just temporarily.

Portions of a public cloud can be carved out for the exclusive use of a single client, creating a virtual private datacenter. Rather than being limited to deploying virtual machine images in a public cloud, a virtual private datacenter gives customers greater visibility into its infrastructure. Now anyone can manipulate not just virtual

machine images, but also servers, storage systems, network devices, and network topology, deployed at a collocation facility.

Creating a virtual private datacenter with all components located in the same facility helps to lessen the issue of data locality because bandwidth is plentiful and typically free when connecting resources within the same facility.

1.2.2. Private Cloud

Private clouds are built for the exclusive use of one client, providing the utmost control over data, security, and quality of service. The company owns the infrastructure and has control over how applications are deployed on it. Private clouds may be deployed in an enterprise datacenter, and they also may be deployed at a collocation facility.

Private clouds can be built and managed by a company's own IT organization or by a cloud provider. In this "hosted private" model, as we installed the Eucalyptus cloud privately within the university, we configure and operate the infrastructure to support a private cloud within the university datacenter.

1.2.3. Hybrid Cloud

Hybrid clouds combine both public and private cloud models. They can help to provide cloud on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to maintain service levels in the face of rapid workload fluctuations. This is most often seen with the use of storage clouds to support Web 2.0 applications. A hybrid cloud also can be used to handle planned workload spikes. Sometimes called "surge computing," a public cloud can be used to perform periodic tasks that can be deployed easily on a public cloud.

Hybrid clouds introduce the complexity of determining how to distribute applications across both a public and private cloud. Among the issues that need to be considered is the relationship between data and processing resources. If the data is small, or the application is stateless, a hybrid cloud can be much more successful than if large amounts of data must be transferred into a public cloud for a small amount of processing.

1.2.4. Community Cloud

One more interesting deployment option is community Cloud [5]. It has an efficient utilization of infrastructure where it is shared by several organizations and supports a specific community. The Community Cloud is as much a social structure as a

technology paradigm, because of the community ownership of the infrastructure. Carrying with it a degree of economic scalability, without which there would be diminished competition and potential stifling of innovation as risked in vendor Clouds.

2.5. Cloud Internals (UEC)

Now we will discuss how the internal working of cloud is. Virtualization is a keystone design technique for all cloud architectures. For our project we have considered the Ubuntu Enterprise Cloud, UEC for short. It utilizes the Eucalyptus to build the cloud on Ubuntu platform. Following are the details of Virtualization technology [1] and other cloud components:

1.3.1. Virtualization

In cloud computing it refers primarily to platform virtualization or the abstraction of physical resources from the people and applications using them. Virtualization allows servers, storage devices, and other hardware to be treated as a pool of resources rather than separate systems, so that these resources can be allocated on demand. In cloud computing, we're interested in techniques such as Para-virtualization, which allows a single server to be treated as multiple virtual servers, and clustering, which allows multiple servers to be treated as a single server. However we have done full virtualization, where the operating systems are deployed directly on the hypervisor (XEN) for creating eucalyptus cloud. Here compute nodes are made on multiple machines, but for the user it looks like single machine with whole set of combined processors and storage. Some specific advantages, including:

Higher utilization rates : Prior to virtualization, server and storage utilization rates in enterprise datacenters typically averaged less than 50%. Through virtualization, workloads can be encapsulated and transferred to idle or underused systems — which means existing systems can be consolidated, so purchases of additional server capacity can be delayed or avoided.

Resource consolidation : Virtualization allows for consolidation of multiple IT resources. Beyond server and storage consolidation, virtualization provides an opportunity to consolidate the systems architecture, application infrastructure, data and databases, interfaces, networks, desktops, and even business processes, resulting in cost savings and greater efficiency.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

Eucalyptus is abbreviated as:

Elastic **U**tility **C**omputing **A**rchitecture **L**inking **Y**our **P**rograms **T**o **U**seful **S**ystems

1.3.3. Node Controller (NC)

A UEC node is a VT enabled server capable of running KVM as the hypervisor. UEC automatically installs KVM when the user chooses to install the UEC node. The VMs running on the hypervisor and controlled by UEC are called \instances". Eucalyptus supports other hypervisors like Xen apart from KVM, but Canonical has chosen KVM as the preferred hypervisor for UEC. Node Controller runs on each node and controls the life cycle of instances running on the node. The NC interacts with the OS and the hypervisor running on the node on one side and the Cluster Controller (CC) on the other side. NC queries the Operating System running on the node to discover the node's physical resources { the number of cores, the size of memory, available disk space and also to learn about the state of VM instances running on the node and propagates this data up to the CC.

Functions:

- Collection of data related to the resource availability and utilization on the node and reporting the data to CC
- Instance life cycle management

1.3.4. Cluster Controller (CC)

CC manages one or more Node Controllers and deploys/manages instances on them. CC also manages the networking for the instances running on the Nodes under certain types of networking modes of Eucalyptus. CC communicates with Cloud Controller (CLC) on one side and NCs on the other side.

Functions:

- To receive requests from CLC to deploy instances
- To decide which NCs to use for deploying the instances on
- To control the virtual network available to the instances
- To collect information about the NCs registered with it and report it to the CLC

1.3.5. Walrus Storage Controller (WS3)

WS3 provides a persistent simple storage service using REST and SOAP APIs compatible with S3 APIs.

Functions:

- Storing the machine images
- Storing snapshots
- Storing and serving files using S3 API

WS3 should be considered as a simple file storage system.

1.3.6. Storage Controller (SC)

SC provides persistent block storage for use by the instances. This is similar to the Elastic Block Storage (EBS) service from AWS.

Functions:

- Creation of persistent EBS devices
- Allowing creation of snapshots of volumes

1.3.7. Cloud Controller (CLC)

The Cloud Controller (CLC) is the front end to the entire cloud infrastructure. CLC provides an EC2/S3 compliant web services interface to the client tools on one side and interacts with the rest of the components of the Eucalyptus infrastructure on the other side. CLC also provides a web interface to users for managing certain aspects of the UEC infrastructure.

Functions:

- Monitor the availability of resources on various components of the cloud infrastructure, including hypervisor nodes that are used to actually provision the instances and the cluster controllers that manage the hypervisor nodes
- Resource arbitration { Deciding which clusters will be used for provisioning the instances
- Monitoring the running instances

In short, CLC has a comprehensive knowledge of the availability and usage of resources in the cloud and the state of the cloud.

Chapter - 3

ISSUES WITH TRADITIONAL CLOUDS

3. ISSUES WITH TRADITIONAL CLOUDS

Clouds offer unprecedented pool of software and hardware resources, which gives businesses a unique ability to handle the temporal variation in their service demands through dynamic provisioning or de-provisioning of capabilities. Actual usage patterns of many enterprise services (business applications) vary over time, most of the time in an unpredictable way. Whenever there is a variation in temporal and spatial locality of workload such as number of concurrent users, total users, and load conditions; each application component must dynamically scale (application service elasticity) to offer good quality of experience to users, and maintain an optimal usage of cloud resources. Cloud-enabling any class of application service would require developing models for service placement, computation, communication, storage with emphasis on important scalability requirements.

3.1. Analysis

Cloud infrastructures are heterogeneous, large-scale, highly dynamic and geographically distributed. Similarly, application services coming in from a large set of public users, brings multiple, independent, and highly distributed software elements embodied in services. Further, the application services arriving at a public cloud also have radically different application characteristics and workload profiles, ranging from the traditional e-Commerce application types, to the newer social networking and collaboration applications, to the enterprise business applications such as CRM and ERP, and to the computing and data intensive type of applications. To appropriately respond to the aforementioned complexity and challenges, application provisioning technologies and approaches should adapt to changing application states and behaviors (leave, join, failure, utilization, availability, workload patterns) of the Cloud computing environments in accordance with high-level guidance specified by the Cloud application developers.

For many enterprises, there is a large amount of IT assets in house, in the form of line of business applications that are unlikely to ever be migrated to the cloud. This may be due to the fact that sensitive data resides in the application and hence cannot reside in a public cloud due to privacy and security issues. Also, increasingly, customer facing departments are creating more web applications to serve customers, which rely on some Cloud platforms, or are services themselves that are deployed on a cloud. As a result, there is a need to look into the scenario and issues related to integration and interoperability between the software on premises and services in the cloud.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

vendors are fulfilling the needs of their respective customers. Now at any given point of a time t, in Cloud-2, if clients ask for more resources dynamically, or resource requirement gets raised significantly, then at time t+α, Cloud-2 cannot meet clients' requirements as it is completely exhausted. Whereas, at the same point of time, other clouds have adequate resources which are free and those will remain underutilized.

For instance, a file hosting company uses a cloud for its storage operation. Now if many users start uploading huge files then the cloud server may face shortage of space resulting into poor QoS. One more practical example is Twitter. Many times we get error message like "Twitter is over capacity. Too many tweets! Please wait a moment and try again" [12]. This happens due to scarce computing resources which are unable to handle unprecedented client requests.

The reason behind this is that no cloud provider has infinite resources. In this situation Cloud-2 cannot ask its frustrated client to go for another cloud provider, as this is going to affect the trust parameter and is going to violet the SLAs. Having enormous pool of resource infrastructure also doesn't seem a good idea, as it involves substantial investment and chances are more that it's going to have adverse effects on the environment as well.

~ ~ ~ * ~ ~ ~

Chapter - 4

Motivation and Related Work

4. MOTIVATION AND RELATED WORK

One way to deal with this could be to observe the simple things that happen in everyday life, like in a grocery shop. Consider what happens when we visit grocery shop for buying the things. If the requested item is not available or not enough then the shopkeeper tries to manage it from some nearby shop and satisfies the customer needs and eases any kind of inconvenience to the customer instead saying straight NO and sending him to another shop.

In general the customer is completely unaware of such things happening in the background.

4.1. Approach

The proposal is about inter-connecting different clouds to address the aforementioned issues. The approach to interconnect cloud could be the use of centralized, hierarchical or peer-to-peer model.

But in case of centralized and hierarchical architecture, there are complications like efficiency bottleneck and single point of failure. We can overcome these problems by choosing the alternative peer-to-peer [3][8] architecture. It also has other advantages like simplicity, distributed system architecture, no centralized control, reliability, no extra energy consumption, no more adverse effect on environment and so on.

In contrast to other networks, nodes are symmetric in function in peer-to-peer architecture. Thus each cloud in such a network is partly a server as well as partly a client and is free to share its own resources.

Figure 4.1 shows the proposed design of connecting clouds with P2P concept. All the clouds participating in this are referred to as peer clouds that help each other by means of sharing the computing resources. Use of this P2P architecture helps the cloud computing to be scalable, sustainable and reliable in real sense.

Now let's see how it works. Say, a client approaches the Cloud-1 for resources and this provider is running out of its infrastructure. At the same time there may be some other clouds whose resources are underutilized and free e.g. of Cloud-2. So Cloud-1 will request Cloud-2 for desired resources on behalf of its client and vice versa. Of course there it's going to be different SLAs between the clouds.

When we say interconnection among clouds then it calls for interoperability [9]. We may achieve something of similar effect by writing high level scripts to orchestrate operations across multiple clouds, discussed in later sections.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

Chapter - 5

OUR PROPOSED ARCHITECTURE

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

III. **Cloud Inter-operability Protocols:** As various cloud providers have different architectures; this module tries to bridge the gap by having higher level scripts that comply with their respective policies. In case of Internet, interoperability foundations [6] [9] were set with the basics of IP addressing, DNS, exchange and routing protocols. We need to examine the corresponding areas and similar technologies, but for cloud computing, like:

A. **Cloud Identity Management :** There is a need of some mechanism which allows clouds communicating over a non-secure network to prove their identity to one another in a secure manner, such as Kerberos [13]. Also, a system which can supply trusted security certificates, such as the X.509 [14] which provides for a public key infrastructure (PKI) for sign-on and Privilege Management. Few modifications in above standard protocols can make it work with clouds.

   Use of IPA [15] seems promising as it is an integrated security information management solution combining an open LDAP directory Server, MIT Kerberos and a X.509 Certificate Authority. IPA provides the functions of:

   - Identity (machine, user, virtual machines, groups, authentication credentials)
   - Policy (configuration settings, access control)
   - Audit (events, logs, analysis thereof)

B. **VM Mobility:** VM Mobility is a feature which allows a running system to be moved from one node to another. The metadata which specifies the system image is very crucial for VM mobility. For this, we can use Open Virtualization Format (OVF) [16]. It is portable and flexible format for the packaging and distribution of VMs over clouds.

C. **Security:** To protect global Cloud exchange for trading services we need to use VPN or SSH Tunnel mechanisms between the peer clouds which ensures that data and the images keep their integrity in the clouds. This issue is also addressed here in [10].

D. **SLA management:** When the clouds exchange the resources/services, there has to be some agreement on provision of services among peer clouds. At given time, one cloud can be considered as the client of another cloud, so is the SLA. This will also resolve the implications concerning SLA fulfillment for a cloud customer when the provider migrates its service to another cloud.

IV. **Resource Advertising:** As this is a kind of business model, every cloud provider participating in this architecture will advertise the extra resources which are free, at particular point of time. As it's in the form of simple XML file, sent at regular intervals, it's not going to add any significant transmission overhead.

V. **Resource Catalogue:** The cloud which needs help from others, will broadcast a query for resources. The other clouds which are ready for this would reply along with SLAs. All these gathered options are kept in a directory called as Catalogue.

VI. **Resource Selection:** After the catalogue generation, the cloud can select an appropriate cloud provider from the catalogue to satisfy its dynamic needs. This selection is based on many parameters like SLAs, trust, pricing, vicinity, past records, etc. The auction model can be used to select cost effective provider(s).

VII. **Resource Binding and Allocation:** Next to the selection of well suited resource provider(s), this module will take care of binding of the client of requesting cloud to the pointers of the peer cloud service provider. This will also take care of incoming requests, when the cloud is acting as Server. After binding, all the resources need to be allocated through a secure connection between the clients of needy cloud to the node controller of wealthy cloud (having excessive resources).

In order to have VM manageability among clouds, we focused on the libvirt project [17]. Libvirt supports features such as remote management using encryption and X.509 certificates, remote management authenticating with Kerberos, discovery using DNS, and management of virtual machines, virtual networks and storage. An Integration of the N2N (network to network) VPN [11] concept may prove very useful. Actually, N2N is a layer-2 over layer-3 P2P virtual private network (VPN) application allowing users to exploit properties typical of P2P applications at the network level instead of application level. This means that users can gain unrestricted IP visibility and be reachable with the same address regardless of their current network environment.

5.4. Peering API in action

We can observe the exact sequence of actions taking place between Requester's Peering API and Providers' Peering API, as shown in Figure 5.3 [21][22]. This figure shows the request-response kind of communication between the needy cloud and multiple wealthy clouds.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

[4] From the catalogue, appropriate cloud provider(s) is selected on the basis of many parameters like trust, past records, QoS, vicinity, pricing, SLAs, etc.

[5] The resource request(s) is sent to the selected cloud provider(s). SLAs and interoperability protocols are negotiated. If everything is OK then acknowledgement is sent. This resembles the standard handshaking protocol.

[6] Thereafter is the job of binding the client of the requesting cloud (needy) to the node controllers of the resource provider cloud(s) (wealthy).

[7] Now it is the time to send the requests to the cloud service provider(s) for resource allocation, as and when needed.

[8] Based on the SLAs, the requests are routed by the resource allocation module in peering API and based on implementation strategy, as discussed, the images are deployed on node controllers and respective services are given to the intended client.

[9] The modules in peering API play vice versa role when direction of provision of services is reversed (when client becomes server and server becomes client).

[10] This accomplishes the target of seamless provision of cloud services using P2P architecture.

5.5. Use of REST

One of the ways to implement the peering API is the use of Representational State Transfer (REST) [18]. It is a way to access resources using simple remote software architecture. The Web (World Wide Web) is a live and classical application of the REST architecture.

Why REST?

Since REST [18] uses standard HTTP it is much simpler in just about every way. Creating specifications, developing APIs, the documentation, etc. are much easier to understand and there aren't very many things that REST doesn't do easier/better than SOAP. REST permits many different data formats whereas SOAP only permits XML. While this may seem like it adds complexity to REST because you need to handle multiple formats, in my experience it has actually been quite beneficial.

REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached. It's worth mentioning [19] that Yahoo uses REST for all their services including Flickr and del.ici.ous. Amazon and EBay provide both, though Amazon's internal usage is nearly all REST source. Google used to provide only SOAP for all their services, but in 2006 they deprecated in favor of REST [20].

~ ~ ~ * ~ ~ ~

Chapter - 6

IMPLEMENTATION DETAILS

6. IMPLEMENTATION DETAILS

Our project describes how to interconnect the peer clouds and how the communication would take place among them. As one can see easily that this is a top level idea and to implement that, it needs multiple cloud providers having different underlying architectures, standardized way of message passing e.g. in XML, incorporation of numerous protocols, VM mobility, cloud interoperability, SLA management, taking care of security issues and the list goes on.

6.1. Simulating the Model

So we tried to simulate the model using simpler resources. We setup two UEC clouds using four PCs. (To setup single cloud, one PC for CLC + CC and another one for NC) Interoperability is no big deal in case of two UEC clouds, as they have identical underlying architecture. So to have effect of multiple clouds, we simulated clouds by means of XML files. That means each XML file is used to describe specific cloud provider. This specification gives details of following parameters :

- Name of Cloud
- SLA (guaranteed uptime, penalties in case of violation, etc.)
- Capacity (in terms of computing speed, data storage, networking devices, primary and cache memory, etc.)
- Provision of Services (IaaS, PaaS, SaaS)
- VM images (OS details, x32/x64, etc.)
- Availability
- Some other necessary factors

Now we use the Apache server and PHP scripting to simulate the operation between the clouds. Say server hosts the XML files (advertising) and a parser that get all the details of cloud and shows it to the other cloud providers, if resources are in excess. Cloud identity verification is done with IPA (already discussed). Now the cloud provider that needs the extra resources, reads the XML files and stores the configuration in a directory (Catalogue). Now by means of PHP scripting, the proper cloud provider is selected based on various parameters. After signing of SLAs among the cloud providers, services are exchanged. Later on agreement, access credentials and service pointers are exchanged. For VM mobility, standardized way is utilized e.g. OVF. All this communication is secured with the use of SSH or vTunnel mechanism.

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

UNIVERSITY OF HYDERABAD
सा विद्या या विमुक्तये

Chapter - 7

CONCLUSION AND FUTURE DIRECTIONS

7. CONCLUSION AND FUTURE DIRECTIONS

7.1. Conclusion

Interconnecting multiple clouds for sharing resources is the hot topic nowadays. This work leverages Peer-to-Peer (P2P) networks to speed up the cost-effective provisioning of cloud infrastructure. It exploits the underutilized computing resources in cloud world and achieves the seamless provision of services even in the event of large fluctuations in computing load that cannot be handled by a single cloud system. This new dimension of computing supports creation of collaborative services and innovations along with backup and recovery management.

In order achieve this, we firstly identified and logically separated the domains, characterizing a layered - 'peering API'. We also examined the critical issues like interoperability, standardization, SLA management, resource mobility and use of REST architecture to implement peering API. Green Cloud computing is envisioned to achieve not only efficient processing and utilization of computing infrastructure, but also minimization of energy consumption, as we are utilizing already provisioned infrastructure. In fact, this may prove to be the basic design for future of cloud computing.

7.2 Future Work

We have given a full-fledged architectural design for solving the problem of scalability and addressed much of the interoperability issues. Our future work will focus on having exact working model of this proposal, i.e. doing practical with the heterogeneous clouds and their different protocols. We will also bridge the gap between those clouds by converting respective APIs to standardized API. In case of this cloud architecture, the customer data gets exposed to the other cloud providers, we will work to minimize the security concerns involved. Also, a client can use its cloud as a proxy to use another prohibited cloud , as client identity is hidden from other peer clouds. We will also pay attention to clear this problem. We will continue this work and implement these ideas to contribute to a basic design paradigm for future of cloud computing.

REFERENCES

REFERENCES

[1] Hurwitz, J., Bloor, R., Kaufman, M., Halper, F., *Cloud Computing for Dummies*, Wiley Publishing, Inc., NJ, 2009

[2] Johnson, D., Murari, K., Raju M., Suseendran, RB., Girikumar Y., *Eucalyptus Beginner's Guide – UEC Edition*, CSS Corp. Pvt. Ltd., 2010

[3] Gupta, A., Awasthi, L., "Peer enterprises: A viable alternative to Cloud computing", Internet *Multimedia Services Architecture and Applications (IMSAA), IEEE International Conference on, 2009*

[4] Parameswaran, AV., and Chaddha, A., "Cloud Interoperability and Standardization", *SETLabs Briefings, VOL 7, 2009*

[5] Marinos, A., Briscoe, G., "Community Cloud Computing", *CloudCom '09 Proceedings of the 1st International Conference on Cloud Computing, 2009*

[6] Assunção, M., Buyya, R., Venugopal, S., "InterGrid: A Case for Internetworking Islands of Grids", *Concurrency and Computation: Practice and Experience (CCPE), Volume 20, Issue 8, pp. 997-1024, Wiley Press, New York, 2008*

[7] BitTorrent, *http://www.bittorrent.com*

[8] Peer to Peer, *http://en.wikipedia.org/wiki/Peer-to-peer*

[9] Bernstein, D., "Keynote 2: The InterCloud: Cloud Interoperability at Internet Scale", *Sixth IFIP International Conference on Network and Parallel Computing, 2009*

[10] Vouk, M.A. "Cloud computing-Issues, research and implementations", *30th International Conference on Information Technology Interfaces (ITI), June 2008*

[11] Deri, l., Andrews, R., "N2N: A Layer Two Peer-to-Peer VPN", at *http://www.ntop.org/n2n/ and in IFIP International Federation for Information Processing, 2008*

[12] *Twitter is Over Capacity*, at http://www.thepicky.com/internet/why-twitter-is-over-capacity-the-truth/

[13] *The Kerberos Network Authentication Service,* and related other RFCs at http://tools.ietf.org/html/rfc4120

[14] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* and related other RFCs at http://tools.ietf.org/html/rfc3280

[15] *The Free IPA Project* at http://freeipa.org

[16] *Virtualization Management (VMAN) Initiative,* Distributed Management Task Force, Inc. at http://www.dmtf.org/standards/mgmt/vman/

[17] *Libvirt virtualization API project*, at http://libvirt.org/

[18] Han, H., Kim, S., Jung, H., Yeom, HY., Yoon, C., Park, J., Lee, Y., "A RESTful Approach to the Management of Cloud Infrastructure", *IEEE International Conference on Cloud Computing, 2009*

[19] *SOAP v/s REST* at http://spf13.com/post/soap-vs-rest

[20] *Google SOAP - No Longer Available*, at http://code.google.com/apis/soapsearch/

[21] Paper communicated :

Snehal Masne, Rajeev Wankar, C.R. Rao and Arun Agarwal, "Seamless Provision of Cloud Services Using Peer-to-Peer (P2P) Architecture", *The International Conference on Parallel Processing, under topic of Cloud Computing, 2011*

[22] Discussion Forums : [username : snehalmasne]

http://ubuntuforums.org/

http://open.eucalyptus.com/forums/eucalyptus-community-cloud

http://serverfault.com/

http://groups.google.ca/group/cloud-computing

~ ~ ~ * ~ ~ ~