Self Adaptive, Energy Conserving: Cloud Storage System

[1.] Rahul S Bhoi , [2.] Arnav Ahire , [3.] Kunal Wattamwar , [4.] Shailesh Shinde, [5.] Prof. S. P. Kosbatwar
[1,2,3,4,5.] Smt.Kashibai Navale College Of Engineering,
University of Pune

***Abstract*— The objective of this research is to present an energy-conserving, self-adaptive cloud storage system. We are building a private cloud using OpenNebula which is widely used for building public, private and hybrid cloud. We are using IaaS (Infrastructure as a service) to deploy operating system to the client .A resource can be shared by single PC or by many PCs & charges can be applied according to the usage of that resource. The data-centre handles the load of the system. It contains other sub controller such as client controller & server controller. They manage the clients and main server's load effectively. This private cloud technique is used in saving a lot of memory which would have been wasted if the same operating system had to be installed on every other PC separately which indeed saves a lot of expenditure and thus making sharing cost effective.**

I. Introduction

Before the advent of cloud computing, traditional computing consisted of buying the resources individually for separate computers in order to use them. The resources which consist of both software and hardware are pretty expensive to buy. When these resources were required on a number of computers it was pretty difficult to buy them individually for every PC and it would prove out to be quite expensive. Ex: In early days of computing, if many computers needed to give a print command to the printer, printer being expensive was either shared between them on one by one basis or each computer was assigned a separate printer. Both the options were inefficient as sharing on one by one basis made the sharing time consuming and assigning separate printers made the sharing more expensive. Cloud computing provides an effective solution to such problem. It allows sharing of resources without any additional hardware or software cost. A resource can be shared remotely using a single PC by many other PCs and charges can be applied accordingly depending on the usage of that resource.

Cloud computing is becoming an important concept in IT world. In the past 10 years the number of users have increased tremendously which has caused services provide via the network to be extremely agile in the terms of availability and load handling. These needs are met via cloud computing. According to U.S. 'National Institute Of Standards & Technology (NIST) the cloud computing is the way of providing network access to a shared pool of computing resources, for instance applications, services as storage. The resources can be provisioned in an on-demand manner & require minim al providers management effort. It provides different services to organization through the internet. Services can be delivered to user wherever and whenever he needs. Services provided by cloud computing are classified as:

A. *Software as a Service (SaaS):*

In this, software application is deployed as a service. Applications to be used are centrally hosted on the cloud. End users can use it without installing it on their computers. User has to pay only for service which has been used.

B. *Platform as a service (PaaS):*

It is the set of tools designed to make coding and deploy applications. It provides platform for developers to create their applications. PaaS provides servers for softwares. Consumers can control deployed applications.

C. *Infrastructure as a Service (IaaS):*

It provides individual servers, disk drives, computing resources; private net etc. 'Pay for what you use' model is applied here. It generally includes multiple users on a single piece of network.

Deployment Models:

- **Public Cloud:** This cloud infrastructure is available to the general public over the internet. It is owned by cloud provider.
- **Private Cloud:** This cloud infrastructure is mostly used in organizations for their internal users only. It can be operated by organization or external cloud provider.
- **Hybrid Cloud:** It is the combination of two or more infrastructures like public, private or community.Advantages of both infrastructures is added to provide better reliability of service.
- **Community:** This is available for users from multiple organizations. It is operated by the organization or a third party.

We mainly focus on IaaS cloud infrastructure which enables on-demand provisioning of resources in the form of Virtual Machines. In our case we are planning to share operating system (OS) as a resource so that other PCs running on the other OS can use different OS according to their need. This would help in saving a lot of memory which would have been wasted if the same OS had to be installed on every other

PC separately which indeed saves a lot of expenditure and thus making sharing cost effective.

II. LITERATURE SURVEY

The history of cloud computing can be traced back to the 1950s when large scale mainframe computers became available in the academia and corporations, accessible via thin clients/mainfranes known as 'static terminals' basically used for communication but they had no internal processing capabilities. This lead to the concept of multitasking which coined another term called time sharing.

In 1990s, telecommunications companies, who previously offered primarily dedicated point-to-point data circuits, began offering VPN (Virtual Private Network) services with comparable quality of service, but at a lower cost. By switching traffic as they saw fit to balance server use, they could use overall network bandwidth more effectively. After that in early 2000s 'dot-com' and 'Amazon' became the pioneers in development of cloud computing by modernizing their data-centres to meet the needs of this new architecture. Their contribution was later embellished by contributions of 'Eucalyptus' platform, 'OpenNebula' and 'IBM' in development of Hybrid cloud and deployment of their respective services. Efforts were made to achieve 'quality of service'.

III. KEY CONCEPTS

A. OpenNebula:

It is an open source toolkit for creating your own IaaS private, public or hybrid cloud environment. It can interface with multiple data-centre services. OpenNebula orchestrates storage, network, virtualization, and monitoring & security technologies to deploy multi-tier services as virtual machines on distributed infrastructures according to the allocation policies. OpenNebula automates the virtual machine setup (preparing disk images & networking) regardless of type of hypervisor (i.e. Xen, KVM, VMware). This cloud toolkit manages virtual machines themselves. It allows support to hybrid cloud.

- **Architecture:**

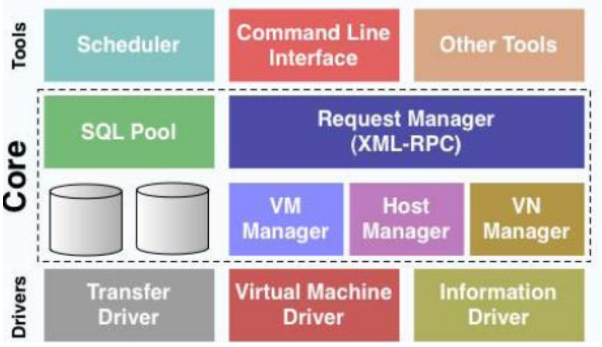

Figure 3.1 OpenNebula Architecture

Architecture is divided into 3 layers:

1. **Tools:** It contains tools distributed with OpenNebula such as CLI, scheduler, the libvirt API implementation etc. A separate scheduler component makes virtual machine placement decisions. It has access to all requests that OpenNebula receives & based on these requests, keeps track of current & future allocations, creating & updating a resource schedule and sending appropriate deployment commands to OpenNebula core. The default scheduler provides a rank scheduling algorithm according to which it places virtual machines on physical resources.
2. **Core:** It contains the components that controls & monitor virtual machines, virtual networks, storage and hosts. By invoking a suitable driver, virtual machine management can be done.
3. **Drivers:** The layer is responsible for interacting with middleware layer such as virtualization hypervisor, file transfer mechanisms. It is designed to provide different virtualization technologies into core.

B. Virtualization:

Virtualization is the process of creating virtual hardware from physical platform so that resources are pooled and can be accessible when it is requested. It is a key technology for resource sharing in IaaS cloud infrastructures. Virtualization lets hardware imitate multiple hardware's through a virtual machine monitor & each virtual machine functions as an individual unit. Virtual machine consists of memories, CPU, and so on. It can run any operating system as guest operating system without affecting other virtual machines. Virtualization enables following cloud computing attributes:

i. **Scalable & elastic:** Services can be scaled on demand; resources can be added or removed as per the requirement.
ii. **Resource Sharing:** Resources can be shared so that η is increased.
iii. **Metered by Use:** Services are build based on usage metrics.

There are four ways to achieve virtualization. Those are hardware emulation, full virtualization, paravirtualization & operating system virtualization.

C. Hypervisors:

Virtualization is provided by applications called hypervisors or Virtual Machine Monitor (VMM). It provides resource to the virtual machines. Virtual machine is an application that emulates the physical machine & has all its characteristics. There are two types of hypervisors: Type1 & Type2. Type 1 runs directly on a hardware platform, for instance Xen, KVM or VMware. Type2 requires an installed Operating System (i.e. host OS) to run. Ex.: VMware Workstation, Virtual Box.

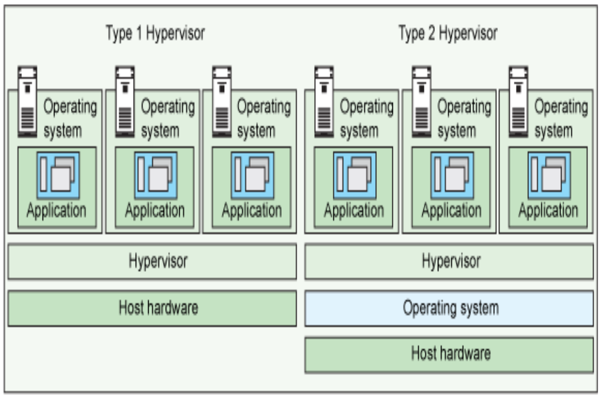

Figure 3.3 Hypervisor Types

OpenNebula supports Xen, KVM, and VMware hypervisors. For our project we are using KVM hypervisor.

- **KVM Hypervisor:**
  Kernel based Virtual Machine (KVM) is a hypervisor for Linux systems. KVM provides full virtualization technique. KVM is used because it is very simple to use.

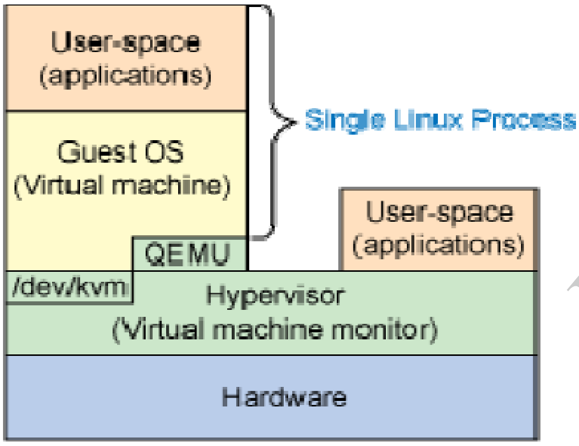

Figure 3.4 KVM Virtualization Components

KVM is implemented by two components:
a kernel module KVM.KO & a processor specific module KVM-intel.KO or KVM-amd.KO. KVM provides the memory virtualization using the /dev/KVM device. KVM/ added a new process mode to Linux called guest operating mode. The mode is used to execute guest OS code without i/o operations. Other two (user & system) modes are available to guest OS. I/O operations are virtualized by a modified QEMU process which runs in a user space & emulates hardware components.

IV. System Model

A. *System Architecture*

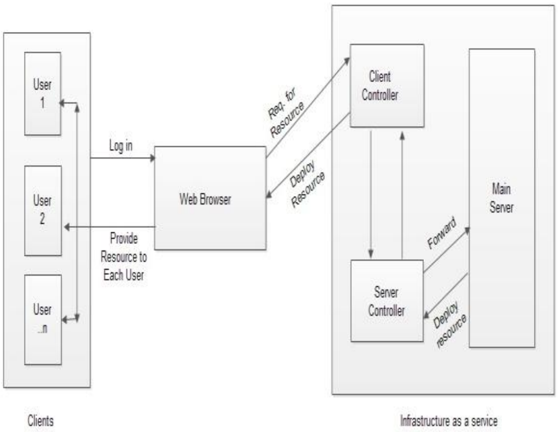

Figure 4.1 System Model

Our proposed architecture contains 2 controllers. One controller is used for client management & another one is used for server management. Client requests through web browser for resource. This request is handled by main server through client & server controller. Above figure shows the proposed architecture of the system.

B. *Cloud Implementation Detail*

Architecture required for building IaaS cloud with OpenNebula.

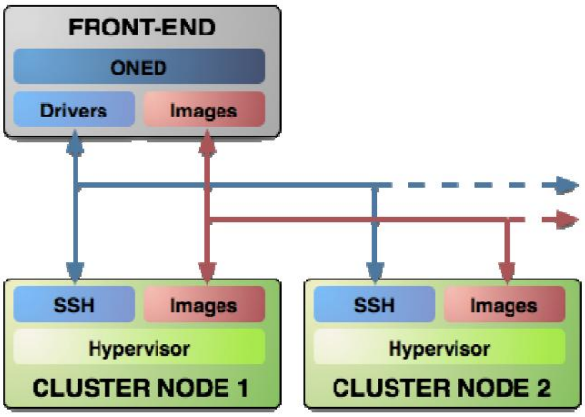

Figure 4.2 IaaS Cloud Architecture

Here 4 PCs are required to build IaaS cloud. It consists of client, client controller, cloud controller, server (front end). In this, client controller and Cloud controller act as a cluster nodes in cloud. Front end is a machine which runs OpenNebula applications. These applications include:

- Oned – a management daemon
- VM-sched – VM scheduler
- Sunstone – Web interface server

Virtual machine images are stored in a image repository that should be accessible by the front end & nodes. The nodes are useful for monitoring & controlling the virtual machines. The hypervisor should be installed on the nodes. They do not have OpenNebula application installed.

All operations are performed via secure shell (SSH) access generated during installation & done internally by OpenNebula services.

C. *Mathematical Model*

SET THEORY ANALYSIS

- Let 'S' be the 'Self Adaptive energy conserving cloud storage system.'
  S= {………………..}
  It consists of 4 modules:
  S1= Client Module
  S2= Client Controller Module
  S3= Server Controller Module
  S4= Main Server Module
  S= {S1, S2, S3, S4}

- Let X denotes the inputs
  Inputs= {X1, X2, X3…Xn}
  X1= Input Message for login
  X2= Input message for resource request
  X3= Input message for resource deallocation

- Let Y denote the outputs
  Outputs= {Y1, Y2, Y3…Yn}
  Y1= Output Message for successful login
  Y2= Output message for resource allocation
  Y3= Output message for resource deallocation

ACTIVITIES

1. Client Module
   S1= {login, password, request}
   Client enters his login ID & password
   If successful then client controller prompts a successful login to client and grants permission to access the resource by accepting client request else prompts an unsuccessful login.
   C-> {C | C is a client ^ C Є (1, 2, 3…n)}
   CC-> {CC | CC is a client controller ^ CC=1}
   ¥ C Ǝ! CC (For every client there exists only
   One universal client controller)

2. Client Controller Module
   S2= {forward request, forward response}
   Forwards the request of client to the server controller and when the resource is granted, forwards the response to the client.
   R-> {R | R is a request}
   SC -> {SC | SC is a server controller ^ SC=1}
   ¥ Σ R(i) ϶ i {1, 2….n} -> SC (All requests are passed from client controller to client)

3. Server Controller Module
   S3= {Forward request, forward response}
   Forwards the request of client controller to the server and when the resource is granted, forwards the response to the client controller.
   R -> {R | R is a request}
   S -> {S | S is a server ^ S=1}
   ¥ Σ R(i) ϶ i {1, 2….n} -> S (All requests are passed from server controller to server)

4. Main Server Module
   S4= {access request, process request, respond}
   Server will accept the request from server controller and depending on number and order of requests it will create virtual machines for clients and deploy services over them.
   V-> {V | V is a virtual machine ^ V -> Σ R(i)}
   ¥ C(i) Ǝ S(i) ϶ [¥ C Ǝ! V.(V(i)=R(i))]
   (For every client there exists a server such that for every client a virtual machine is created for every request)

V. Conclusion and Future Work

There are still open research challenges, though there is lot of revolution in cloud computing. This paper is mainly focused on cost effective resource sharing. Rank scheduling algorithm used in the scheduler facilitates efficient use of memory making the system energy efficient as well as self adaptive.

In the future we are planning to extend this project to share multiple resources using multiple type of services provided by the cloud. Furthermore we will extend it for hybrid cloud.

VI. Acknowledgement

With deep gratitude we acknowledge the support of our guide, teachers, other staff members and friends for supplying us valuable information regarding our project.

References

[1] Gaurav Raj, Ankit Nischal "Efficient Resource Allocation in Resource provisioning policies over Resource Cloud Communication Paradigm ", International Journal on colud computing services & architecture (IJCCSA),Vol.2,No.3, June 2012.

[2] Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy "Energy-aware resource allocation heuristics for efficient management of datacentres for cloud computing."

[3] Ronak Patel, Sanjay Patel "Survey on Resource Allocation Strategies in Cloud Computing", International Journal of Engineering Research & Technology (IJERT), ISSM: 2278-0181, Vol. 2, Issue 2, Feb 2013.

[4] Maciej Byczuk"A private cloud based on the OpenNebula toolkit"

[5] Yong Zhao, Yanzhe Zhang, Wenhong Tian, Ruini Xue, Cui Lin "Designing and Developing a Scientific Computing Cloud Platform", 2012 ACM/IEEE 13th International Conference on Grid Computing.

[6] Thomas J. Bittman,"Server Virtualization: One Path That Leads to Cloud Computing", Gartner RAS Core Research Note G00171730,29 October 2009, RA3 12012010.

[7] OpenNebula , www.opennebula.org , 2012.

[8] http://en.wikipedia.org