Dynamic Private Small Cloud for Linux Thin Clients

Jahnavi G Reddy[1], Shubhashree S[2], Shruti Shetty[3], Sanjeetha R[4]
Dept. of Computer Science and Engineering
M S Ramaiah Institute of Technology
Bangalore, India
[1]janu.g.reddy@gmail.com, [2]shubhashree.s@gmail.com, [3]shruti.shetty25@gmail.com, [4]gsanju.prakash@gmail.com

Abstract **— Energy consumption for high performance computing is enormous. This can be substantially reduced using Cloud Computing which provides low cost, energy efficient and ubiquitous computing. But Cloud Computing cannot work efficiently with limited resources.**

We aim to build a Private Small Cloud (PSC) that is based on small clusters, virtualization and graphics processor which can handle large amounts of data with low cost, high efficiency and reliability using limited resources. Using Cloud Computing for low power PC cluster architecture and low power client devices can reduce energy consumption and carbon emission. Virtual machine server run multiple operating systems simultaneously on a single computer and supports multiple heterogeneous applications running on a number of client devices. General purpose graphics processing unit (GPGPU) makes use of parallel processing and multithreading. The cloud is designed to work with Linux thin clients using wired connection.

Keywords- private small-cloud computing; Ubuntu Enterprise Server; linux-based embedded devices; JamVM virtual machine

I. INTRODUCTION

Recently, cloud computing has become a popular computing paradigm in which virtualized and scalable resources are provided as services over the Internet. However the running of large scale computing and data centers generally requires a large amount of energy. In fact, enormous energy is wasted due to idle resources. Moreover, issues of security and system flexibility arise while using a public cloud. These can be solved by managing our own cloud network which is not only advantageous for the above reasons but also reduces cost.

We propose to solve this, by setting up our own Private Small Cloud (PSC) which is based on three concepts: small clusters, virtualization and general graphics processor [1]. Cluster computing is preferred over grid computing due to fault tolerance, robustness, rapid recovery and high performance computing.

Next, a virtual machine server is a service designed to run multiple operating systems simultaneously on a single computer and supports multiple heterogeneous applications running on a number of client devices. A single machine cannot fit for all the tasks at the same time, therefore the virtual machine server is deployed to reduce power consumption, save time, cost and improve execution.
Finally, the use of a general purpose graphics processing unit (GPGPU) can be realized for stream processing rather than a general processor. Stream processing is one of the paradigms of compilation of a program and is related to SIMD computer programming for which GPGPU has easier use of a limited form of parallel processing and multithreading.

Walrus storage controller in conjunction with cloud controller can be used to store large amounts of data. In many applications, embedded devices often require huge computing power and storage space, cloud computing services can be used to achieve this goal. A web interface can be deployed at the client side to use the cloud. The working of the private cloud can be tested using a database application.

II. BACKGROUND

Deploying a cloud structure generally needs the following softwares: Xen, OpenNebula, Eucalyptus, and Euca2ools. An open source Ubuntu Enterprise Server with the option Ubuntu Enterprise Cloud is utilized to build the private small-cloud. The current Ubuntu Enterprise Cloud 10.04 includes Xen, VMGL, Open Nebula, and Eucalyptus and other packages. Through these we can focus on installing the back-end cluster controllers and cloud controller in order to build a private small-cloud.

Due to its low cost and ease of customization, Linux is often used in embedded systems. Android—based on a modified version of the Linux kernel—has become a major competitor of Nokia's older Symbian OS, found in many smart phones. Linux is used in embedded systems like mobile phones, PDAs, set top boxes etc. Cell phones and PDAs running Linux on open-source platforms became more common from 2007; examples include the Nokia N810 and the Motorola ROKR E8.

JamVM is an open source Java Virtual Machine (JVM) developed to be extremely small compared with other virtual machines (VMs) while conforming to the Java virtual machine specification version 2 (blue book)[10]. JamVM can be configured to use the GNU Classpath or the OpenJDK Java class library and recent versions support object finalization, Soft/Weak/Phantom References, the Java Native Interface (JNI) and the Reflection API. The compacting garbage collector can run either synchronously or asynchronously within its own thread. JamVM currently supports the CPUs: AMD64, ARM, i80486, MIPS, PowerPC and SPARC.The OpenJDK compatible version of JamVM is supported by IcedTea, and IcedTea packages of JamVM are included in

both Debian and Ubuntu. This enables JamVM to be installed as an alternative Java Virtual Machine to hotspot when using OpenJDK. When using Ubuntu on ARM, JamVM is the WS3 or Walrus Storage controller provides a persistent simple storage service using REST and SOAP APIs compatible with S3 APIs. Some of the functions of Walrus are: storing the machine images, storing snapshots, storing and serving files using S3 API.

Xen is a virtual-machine monitor providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.The University of Cambridge Computer Laboratory developed the first versions of Xen. Since 2010, the Xen community develops and maintains Xen as free software, licensed under the GNU General Public License (GPLv2). In Xen systems the Xen hypervisor is the lowest and most privileged software layer. This layer supports one or more guest operating systems, scheduled on the physical CPUs. The first guest operating system, called in Xen terminology *domain 0* (dom0) is executed automatically when the hypervisor boots and receives special management privileges and direct access to all physical hardware by default. The system administrator can log into dom0 in order to manage any additional guest operating systems, called user domains (*domU*) in Xen terminology.

OpenNebula is an open-source cloud computing toolkit for managing heterogeneous distributed data center infrastructures [5]. The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid IaaS (Infrastructure as a Service) clouds. OpenNebula orchestrates storage, network, virtualization, monitoring, and security technologies to deploy multi-tier services (e.g. compute clusters) as virtual machines on distributed infrastructures, combining both data center resources and remote cloud resources, according to allocation policies. OpenNebula is used by a variety of organizations, including hosting providers, telecom operators, IT services providers, supercomputing centers, research labs, and international research projects.

Eucalyptus is a software platform for the implementation of private cloud computing on computer clusters [6]. There is an open-core enterprise edition and an open-source edition. Currently, it exports a user-facing interface that is compatible with the Amazon EC2 and S3 services but the platform is modularized so that it can support a set of different interfaces simultaneously. The development of Eucalyptus software is sponsored by Eucalyptus Systems, a venture-backed start-up. Eucalyptus works with most currently available Linux distributions including Ubuntu, Red Hat Enterprise Linux (RHEL), CentOS, SUSE Linux Enterprise Server (SLES) and Fedora. It can also host Microsoft Windows images. Similarly Eucalyptus can use a variety of virtualization technologies including VMware, Xen and KVM hypervisors to implement the cloud abstractions it supports. Eucalyptus is an acronym for “Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems”.

Euca2ools are command-line tools for interacting with Web services that export a REST/Query-based API compatible with Amazon EC2 and S3 services [3]. The tools can be used with both Amazon's services and with installations of the Eucalyptus open-source cloud-computing infrastructure. The tools were inspired by command-line tools distributed by Amazon (api-tools and ami-tools) and largely accept the same options and environment variables. However, these tools were implemented from scratch in Python, relying on the Boto library and M2Crypto toolkit.

The components of the Private Small Cloud are as follows:

A. Virtual Machine (VM):

A single machine cannot fit in all the tasks at the same time, therefore the virtual machine server is a service designed to run multiple operating systems simultaneously on a single computer and supports multiple heterogeneous applications running on a number of client devices.

B. Node Controller (NC):

The NC (through the functionality of a hypervisor) controls VM activities, including the execution, inspection and termination of VM instances. Prior to installing a node controller an ISO file booting system with English language choice should be done as command line style is the input mode for the node controller.

C. Cluster Controller (CC):

The CC controls the execution of virtual machines (VMs) running on the nodes and manages the virtual networking between VMs and external users. It manages all the clusters.

D. Storage Controller (SC):

The SC provides block level network storage that can be dynamically attached by VMs.Walrus can be used to provide this functionality.

E. Cloud Controller (CLC):

The CLC is responsible for exposing and managing the underlying virtualized resources (machines, network and storage) via user-facing APIs via Web interfaces. It interacts with the cluster and storage controller.

III. METHOD

In order to deploy a minimum of cloud structure, at least two dedicated systems are needed [2]. One will be used as a cloud controller (clc), and contains the entire back-end cluster controller (cc), storage server Walrus, and the storage controller (sc). This host needs fast disks and a few fast processors to match those disks. Another one is a node controller (nc), used to perform many of the cloud entity.This host takes a lot of capacity with Virtualization Technologies (VT) of the CPU, a large number of CPU computing power,

large memory and fast disk. Building the system in the following steps:

A. Installing virtual machine

In this project we adopt Ubuntu Operating system as the virtual machine as it supports the Ubuntu Enterprise Cloud.

B. Deploying cloud computing architecture

Deploying cloud structure will generally need the softwares: Xen, OpenNebula, Eucalyptus, and Euca2ools. Since system installation needs many steps, manipulation often encounters some errors and the configuration is not easy. This project employs an open source, Ubuntu Enterprise Server Edition, because this version of the Ubuntu has included all of the above packages that are used to deploy cloud structure rapidly and easily.

C. Installing node controller

After the installation of the cloud controller, the node controller is installed using the same ISO booting file as the cloud controller.

D. Setting cloud controller

Back to the cloud controller , commands are executed to find the node controller and examining a link to node controller that is created earlier.

E. Setting cloud user through web interface

Before the user at client side uses the cloud, the client is required to do some of the settings in cloud controller through a web interface called the Login Management Interface: The default account is admin and password admin.

F. Deploying application

Once the Private Small cloud is setup. Webmin is installed on the cloud controller and an RTO database application is deployed. This application can be accessed by all the nodes using the ip address of the cloud controller.

IV. CONCLUSION

We have been able to setup a private small cloud on our own. By building a private small cloud we have observed that it can work efficiently with limited resources. The RTO application deployed on this cloud works faster specially when many users try to access it at the same time.

The future work can include many enhancements to the existing system. The scalability of the network could be increased by building a more complex Private Cloud with more nodes depending on the size of an organization. There could be multiple cluster controllers within the cloud, and multiple node controllers reporting to each cluster controller.

The system developed currently deploys software applications on the cloud. The existing model could be extended to provide Infrastructure as a service and Platform as a service as well.

Another area which could be enhanced is the accessibility of the Private Cloud on wireless mobile devices via 3G network connections. This could be done by utilizing JamVM, GTK + DirectFB, GTK + X11, QT / Embedded.

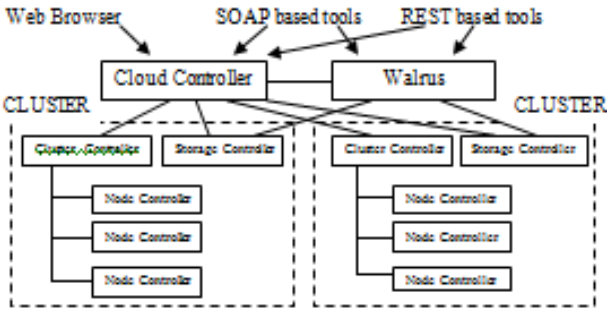

Figure 1. Cloud controller (CLC) architecture.

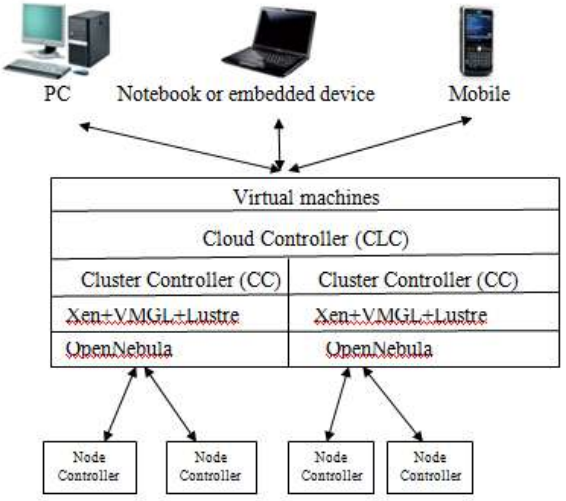

Figure 2. Node controller (nc) structure.

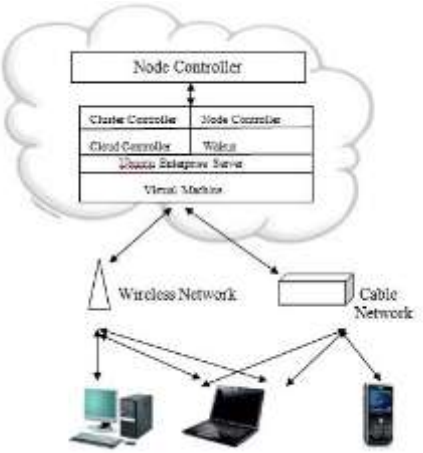

Figure 3: A complete structure of CLC+NC private small cloud computing

Figure 4: RTO application Deployed on the cloud

ACKNOWLEDGMENT

We are grateful to M S Ramaiah Institute of Technology for providing us with all the facilities needed to work on this project. Further, we wish to express our sincere gratitude to our Head of Department,for technically equipping us to successfully complete this project. We take great pleasure in expressing our heartfelt gratitude to all our teachers , for imparting her highly valuable guidance at every stage of the project. We would also like to thank all the teaching and non-teaching staff for their kind co-operation, support and guidance. Last but not least, we wish to avail this opportunity, to thank all our family and friends who have supported us through everything.

REFERENCES

[1] Bao Rong Chang, Hsiu Fen Tsai, Chien-Feng Huang and His-Chung Huang, Private Small-Cloud Computing in Connection with Linux Thin Client, 2010

[2] Ubuntudocumentation https://help.ubuntu.com/community/UEC/CDInstall

[3] Euca2oolsUserGuide,2010 http://open.eucalyptus.com/wiki/Euca2oolsGuide_v1.1

[4] Ubuntu Forums

[5] OpenNebula, 2010. http://www.opennebula.org/

[6] Eucalyptus, 2010. http://open.eucalyptus.com/

[7] Welcome to Apache Hadoop, 2010. http://hadoop.apache.org/

[8] General-Purpose Computation on Graphics Processing Units, 2010. http://gpgpu.org/

[9] Java2Platform,MicroEdition(J2ME),2010. http://www.java.com/zh_TW/download/faq/whatis_j2me.xml

[10] JamVM--AcompactJavaVirtualMachine, 2010.http://jamvm.sourceforge.net/

[11] GNU Classpath , GNU Classpath, Essential Libraries for Java, in2010. http://www.gnu.org/software/classpath/

[12] Ubuntu Enterprise Server , 2010.http://docs.sun.com/app/docs/doc/821-1045/ggfrh?l=zh_TW&a=view