On Construction of Cloud Virtualization with Integration of KVM and OpenNebula*

Chao-Tung Yang, Shao-Feng Wang, and Chi-Jui Liao

Department of Computer Science, Tunghai University, Taichung, 40704, Taiwan ROC
ctyang@thu.edu.tw

Abstract. The goal of this paper is how to build cloud IaaS environment for operation, and integrate KVM and OpenNebula open sources to provide a cloud virtual environment for users. In the user interface part, this work can reduce the complexity of cloud resources accessing for the user part. This paper uses the web interface that is easy to understand, accessible for users in the operations. In the experimental results, this work compares the performance of physical machine and KVM virtual machine, and analyzes the results.

Keywords: Cloud computing, Virtualization, IaaS, KVM, OpenNebula

1 Introduction

Cloud computing is currently a popular topic, but also all the main axis of development in recent years, the main points of infrastructure as a service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), cloud computing is not a new technology; it is a new concept [1, 2]. The early stages of the laboratory started in the creation and development of grid computing cluster and other distributed computing technologies and related issues, for the vigorous development in recent years is also very interested in cloud computing [13, 15].

This paper focused on the cloud computing infrastructure, particularly virtual machines and physical monitoring component [3, 4, 5, 6, 7, 8, 9]. Goal is to achieve a system can provide users apply and use the virtual machine, and can monitor the physical system. The information can be monitored include CPU utilization, disk usage, virtual machine space, memory usage. This system also uses a mechanism for Migration, when a problem occurs, the administrator can shift the user's virtual machine to another physical machine operation, and the user will not feel any abnormalities. Meanwhile, this paper also carried on the system performance test using the KVM [10, 11, 12, 14].

* This study was supported in part by the National Science Council, Taiwan ROC, under grant numbers NSC 100-2218-E-029-004 and NSC 100-2622-E-029-008-CC3
2 System Implementation

Besides managing individual VMs’ life cycles, this study also designs the core to support service deployment. Such services typically include a set of interrelated components (for example, a Web server and database back end) requiring several VMs. Thus, a group of related VMs becomes a first-class entity in OpenNebula. Besides managing the VMs as a unit, the core also handles context information delivery (such as the Web server’s IP address, digital certificates, and software licenses) to the VMs [8].

OpenNebula is a virtual infrastructure engine that enables the dynamic deployment and reallocation of virtual machines in a pool of physical resources. OpenNebula extends the benefits of virtualization platforms from a single physical resource to a pool of resources, decoupling the server, from both the physical infrastructure and the physical location [4]. OpenNebula contains one front end and multiple back ends. OpenNebula orchestrates storage, network, virtualization, monitoring, and security technologies to enable dynamic placement of multi-tier services (groups of interconnected virtual machines) on distributed infrastructures, combining both data center resources and remote cloud resources, according to allocation policies [4].

However, OpenNebula lacks a GUI management tool. Previous works build virtual machines on OpenNebula and implement Web-based management tools. Thus, the system administrator can easily monitor and manage the entire OpenNebula System on our project. OpenNebula is composed of three main components: (1) the OpenNebula Core is a centralized component that manages the life cycle of a VM by performing basic VM operations, and also provides a basic management and monitor interface for the physical hosts (2) the Capacity Manager governs the functionality provided by the OpenNebula core. The capacity manager adjusts VM placement based on a set of predefined policies (3) Virtualizer Access Drivers. To provide an abstraction for the underlying virtualization layer, OpenNebula uses pluggable drivers that expose the basic functionality of the hypervisor [5].

As an Infrastructure as a Service (IaaS) provider, this paper applies an ideas of virtualizes in the cloud system to economize power, web interface and user friendly to manage the virtual machines. Therefore, there are some distinct on framework of cloud; our system architecture is shown in Figure 1. About user friendly, users simply connect to the site through the Internet, and then set their own needs, you can create a virtual machine, the user does not need to know what happened back may need to set any object, they can be consistent with their own Needs of virtual machines. The screen snapshots are shown from Figures 2 to 4 seriously.

According to the previous plan, this paper have established physical machine using KVM virtual machine system and provides a Web interface to manage the virtual machine. Our virtual system was built up with four homogeneous computers; the hardware of these computers is equipped with Intel i7 CPU, eight gigabytes memory, 1TB hard disk, CentOS 6.0 x64 OS, and the network connected to a gigabit switch.
3 Conclusions

This paper implemented a cloud of KVM infrastructure and monitoring website, which offers users to apply for the use and monitoring of VM state, and the main page with easy to understand the type, the user in the application and monitoring, can Ob-
tained through the needs of the most simple steps in order to user friendly. Unlike the past, the usage of Xen as the virtualization technology, this paper tries to use KVM virtualization technology as a major. In addition, this paper also tested live migration and implementation of the efficiency of the test, although there is still a gap from the best performance, but the final results were very satisfactory.

References