KVM Based Virtualization for Low Cost

Prashant H. Band, Seema R. Chaudhary, Radhakrishna Naik

Abstract—In this paper, we propose a Virtual Remote environment to user over a LAN by using open source system and virtualization tools. Today's Desktop processors offer great computation functionality, fast processing power and memory. The aim is to provide low cost computing using open source technology and various free remote desktop environments. We used free Terminal Server [1], [2] to provide desktop, remote desktop tools and KVM (Kernel-based Virtual Machine) hypervisor [3], [4] to create and run virtual machines. Users can get a personalized OS (Operating System) which is the VM (Virtual Machine) running over a Local server. User gets an independent OS through the various graphical desktop sharing systems, which is platform independent. Complete resource utilization is possible because VMs run on Remote Server. As we are using open source OS and tools, this implementation becomes inexpensive. Some solutions like Oracle VM server for SPARC [5], CtrixXEN Desktop [6], and VM ware View [7], [8] are available but they are costly and proprietary. By using this method we can make use of very low configured system as a client system to access VM's desktop for best performance.

Keywords—OS, Virtualization, Hypervisor, KVM, and Remote Desktop.

I. INTRODUCTION

In today's era operating systems are upgrading rapidly, so every time end users need to change or replace software as well as hardware .Every time replacement of new hardware is not feasible solution for end users. But modern computers used by users now a day have a good hardware and powerful computational capacity. These modern computers become outdated if newer versions of operating systems and software's are introduced. Even though these new release in operating system and software takes place, these hardware can perform better computation. If user changes his hardware resources according to software, then it will result in unnecessary wastage of resources and time. These replaced resources have more processing power than required. One solution can be done if user gets remote desktop facility coupled with some older machine, this will result in cost reduction and resource utilization. With this solution user does not work on powerful machine, but share powerful machine resources using thin clients. These thin clients can be any outdated system with low configuration and at least support an operating system which can handle remote desktop client services. This solution saves the time required

for maintenance of machines as we are upgrading the server system only. As we are using outdated system as thin clients we are actually saving the cost required to implement new setup in the organization and better CPU utilization.

Traditional remote desktop solution provides a single user interface for remote terminal with shared environment and desktop.

In some remote desktop solution, powerful servers are used which shares one instance of operating system with multiple users which can work remotely. But this solution faces some problems like, one user can cause harm to other users, as the operating system instance is shared or user gets a session not a personalized desktop.

The cost, efforts and above mentioned problems can be solved by extending remote desktop feature using virtualization. We have combined feature of remote desktop with virtual machine [9] to provide virtualized dedicated remote desktop environment which can be accessed by thin clients on LAN or WAN using any OS or wireless devices. In market some proprietary solutions are already available, which provides same facility, but cannot be owned by small organization and institutions. Our solution is based on open source software and normal desktop computers which act as remote desktop server with increasing CPU utilization and reducing cost. This technique utilizes complete resources hence can be used to improve the performance and reduce the cost or efforts required.

II. BACKGROUND AND RELATED WORK

Modern computers are sufficiently powerful to use virtualization [10], [11], [12] to present the illusion of many VM's (Virtual Machines), each running separate operating system instance. Virtual machines must be isolated from one another; it is not acceptable for the execution of one to adversely affect the performance of another. Virtualization must support multiple operating systems to accommodate heterogeneity. Remote desktop virtualization implementation works on client/server environment. Execution is on remote system which is linked to local client device over a network using remote desktop protocol. All application and data used remain on the remote system where as display, keyboard and mouse at local client device which may be conventional PC/laptops or a thin client device or a tablet, or even smart phones. This implementation is possible with multiple desktop operating systems on a server running with hypervisor [13], [14].

This idea of virtualization using a computer system is useful for resource utilization purpose. In the last decade, VMware software only virtual machine monitor has been quite successful and today VMware view [15]. More recently, the XEN [13] open source hypervisor brought virtualization to the open source world. The new release Redhat enterprise Linux, RHEL6 [16] has decided to migrate to KVM as its supported virtualization platform.

The translation to KVM in RHEL6 comes with several features which either does not exist in XEN or were not implemented in original XEN environment.

In previous work, XEN hypervisor is used to create virtualization environment and LTSP (Linux Terminal Server Project) [2], [17] is used to access Remote Desktop by making some changes in LTSP script [17]. In previous work user can access remote desktop within LAN or outside network. The model consists of two main components [17], one is management server for creating VM on physical machines user to VM mapping and second is a physical machine running VM to provide Remote Virtual Desktop. Only Linux operating system is provided as remote virtual desktop and need some changes if user wants to use Microsoft operating system [17].

III. SYSTEM DESIGN

Here we introduce Ubuntu operating system used as a host operating system and KVM hypervisor as a virtualization environment to runs various operating system (Guest OS). Ubuntu LTSP [1] is used to provide remote desktop as well as used to provide boot desktop computer from the networks where it download the kernel and run them in stateless fashion. This system consists of three main components: HTTP (Hypertext Transfer Protocol) server, LTSP server and Physical (for virtualization). HTTP server is used to create and manage VM's on the Physical System. User can request through any remote desktop client application to access a Virtual Desktop (running VM) as per requirement.

KVM hypervisor is installed on Physical Machine as shown in Fig. 1, which runs user Virtual Machine. The purpose of this LTSP is to provide a login screen through which user can connect to the LTSP server and run entire desktop on the LTSP server to those users who don't have storage. To start and run the VM's when user requests for remote desktop, we have designed scripts and daemons which run on Physical Machines.

The objective of this work is to provide complete independent operating system in the form of Remote Desktop. Functionalities of this Virtual Remote Desktop are very similar to regular desktops available on independent machine. This virtual remote desktop is accessible through any thin client or wireless devices (Smartphone's, tablets). If user wants any proprietary version of operating system as virtual remote desktop, this system is also implemented by using RDP (Remote Desktop Protocol). For cost effective solution we have used Ubuntu as a host operating system as well as guest operating system.
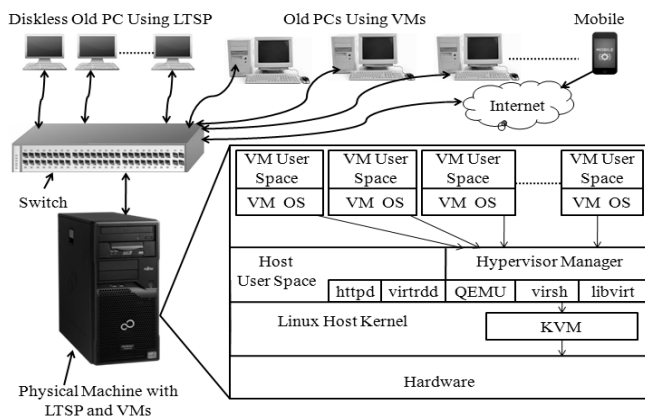

Fig.1. Virtual Remote Desktop

IV. COMPONENTS REQUIRED FOR VIRTUALIZATION

A. Hypervisor: KVM [13] is new hypervisor that has gained popularity in the past few years, but not XEN, which supports paravirtualization as well as hardware assisted virtualization [13]. KVM supports some specific processors like AMDV and Intel VTx processors [13]. KVM is implemented as kernel module which when loaded, converts the kernel into bare metal hypervisor. KVM was designed and implemented after the development and release of hardware assisted virtualization extensions and has been optimized to take a full advantage of these extensions rather than building them in as an afterthought. Due to this, KVM requires Intel VT or AMD-V extensions be present and enabled on host system. AS we are using Intel VT extension system, this choice is suitable for virtualization

B. LTSP: LTSP is used is used to provide remote desktop for Linux users, who are diskless client. LTSP does not require any software or OS to use LTSP remote desktop. It uses only a PXE (Pre Execution Environment) enabled network card [17].

C. Other Remote Access Mechanism: According to the guest OS running on hypervisor, the choice can be made using both guest OS and host OS. If open source solution is required LTSP, VNC (Virtual Network Computing) [19], rdesktop (Remote Desktop Protocol client) [20] can be used. If proprietary solution is used RDP/rdesktop can be used.

V. CONCLUSION AND FUTURE WORK

KVM provides an excellent platform for deploying Virtual Machines to provide a virtualization based Remote Desktop. We are using open source tools, so we expect that there is no need for frequent upgradation of systems and management becomes very easy with virtualization. The complete resource utilization is possible with our work. Via incomprehensive experimentation, our result shows the performance of KVM and Intel VT based systems is much better than old independent systems.

This work is implemented on single system; we are working to combine multiple systems to provide Virtual Remote Desktop, so Load balancing is not implemented.

REFERENCES

1. "Ubuntu LTSP", https://help.ubuntu.com/community /UbuntuLTSP.
2. "Linux Terminal Server Project", www.ltsp.org.
3. www.linux_kvm.org/page/HOWTO.
4. www.linuxnix.com/2013/02.
5. "Running Oracle Real Application Clusters on Oracle SPARCE", An oracle white paper, July 2012.
6. en.wikipedia.org/wiki/Citrix Systems.
7. www.vmware.com/products/view/overview.html.
8. www.vmware.com/in/products/desktop_virtualization/workstation/ overview.html.
9. en.wekipedia.org/wiki/Virtual_machine.
10. en.wekipedia.org/wiki/Virtualization.
11. en.wekipedia.org/wiki/ Desktop_Virtualization.
12. Paul Barham, Boris Dragonic, "Xen and the art of virtualization," in ACM 2003.
13. "A performance Analysis of XEN and KVM hypervisors for hosting XEN Worlds Project" (2011) Charles David Graziano, Graduate Theses & Dissertation paper 12215.
14. en.wikipedia.org/wiki/Hypervisor.
15. en.wikipedia.org/wiki/VMware_view.

16. https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/.
17. "Low Cost Computing Using Virtualization For Remote Desktop", Dhaval Manvar, Mayank Mishra, Anirudha Saho, Computer Science And Engineering Department, Indian Institute of Technology Bombay, 978-1-4673-0298 2012 IEEE.
18. en.wikipedia.org/wiki/X86_virtualization.
19. en.wikipedia.org/wiki/Virtual_Network_Computing.

AUTHOR PROFILE

Mr. Prashant Band received the B.E. degree in Information Technology from Govt. College of Engineering Aurangabad and pursuing M.E. in Software Engineering from M.I.T. Aurangabad.

Assoc. Prof Seema Chaudhary received M.E. degree in Computer Science and Engineering from Govt. College of Engineering Aurangabad. Three International Publications in Data Mining. She is working as Assoc. Prof. in M.I.T. College of Engineering Aurangabad.

Dr. Radhakrishna Naik received PhD degree from S.G.G.S. Nanded. Research topic of PhD is Design and analysis of intelligent real time scheduler: A case study of Greenhouse monitoring system. Papers presented in six conference and six journals. He is working as Professor in M.I.T. College of Engineering Aurangabad.