Deploying Virtualization on Ubuntu using Libvirt and KVM/QEMU Bare Metal Hypervisor

Kishor Bhosale[1], Vidula Bhosale[2]

[1] *Master of Computer Application Department, Finolex Academy of Management and Technology, Ratnagiri, Mumbai University*
[2] *Information Technology Department, Gogate Joglekar College, Ratnagiri, Mumbai University*

***Abstract*— Virtualization empowers us to reinforce IT assets into single unit with the goal that they can be shared over a system and satisfy the contracting assets and developing business needs. It includes and subtracts distinctive registering assets to present or past working situations with various kinds of techniques like equipment and programming accumulation, machine recreation, copying and numerous others. Virtualization is basic piece of distributed computing as it backings various Virtual Machines with various facilitating situations. Cloud stage not just deals with the assets of the cloud foundation additionally benefits client on request necessities and assignment planning on virtual machines. In this paper authors have chosen the bare metal hypervisor KVM/QEMU, libvirt and Ubuntu operating system for the deployment of virtualization. The aspiration of the paper is to provide the qualitative implementation and application of the bare metal hypervisor for the virtualization learning aspirants in the fewer infrastructures. The complete deployment with scripts and screen of the implementation steps has been shown.**

***Keywords*— Virtualization, Virtual Machines, Hypervisor, KVM/QEMU, libvirt.**

I. Introduction

In the course of recent years Information Technology has been encountering quick and very particular errands arranged development. The new patterns in data innovation have made the tremendous test of conveying and overseeing IT framework on a bigger scale. In this sense one of the significant difficulties is scaling up of any current IT framework so as to satisfy the constantly requesting necessities of the present IT prerequisites. To determine these issues IT made the stride ahead and entered in the period of cloud computing. Cloud computing is a domain to enabling invaluable, on-request access of system to a conferred heap of configurable figuring resources (e.g., frameworks, servers, stockpiling, orders, and organizations) that could be immediately provisioned and discharged with insignificant organization effort or organization provider participation [1]. In cloud computing, virtualization assumes an imperative part by rearranging administration and enhancing asset productivity. The greater part of the general population considers virtualization and cloud computing comparative however are very unique in nature and appropriateness. However cloud computing can be spoken to as an administration supplier where virtualization is one of the piece of physical infrastructure [1].

A. Hypervisors

Diverse customizations of hypervisors are accessible in the market for individual and business utilize. The Type 1 (native/ bare metal) hypervisors run straight on the host's gear to control the fittings and guest operating system. A guest operating system, along these lines keeps running on a substitute level over the hypervisor. Though Type 2 (hosted) hypervisors run straightforwardly on host operating system framework structure including the layer of hypervisor and base operating system, the virtual machines keeps running on third layer, above the hardware. The hypervisor conveys the guest operating system with a operating system stage and it additionally deals with the execution of the guest operating system. The significant separations are as per the following:

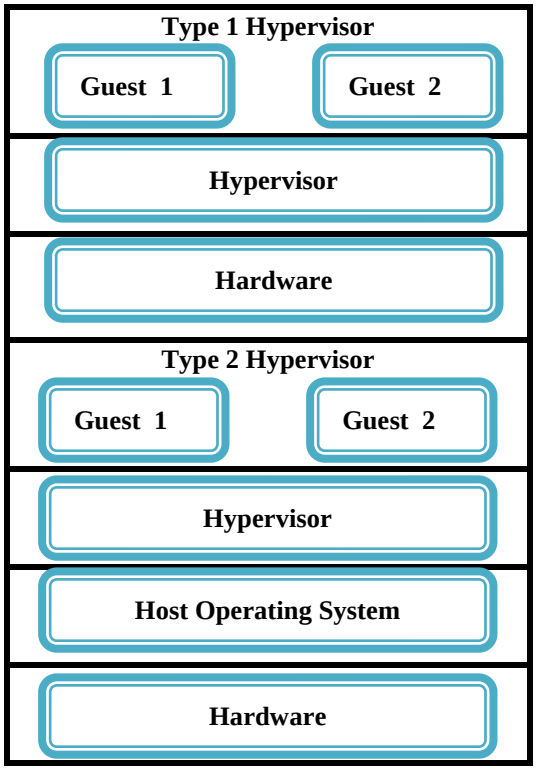

Fig 1: Type of Hypervisors

Numerous occasions of operating system may share the virtualized equipment assets [1]. Google starting late moved its cloud IaaS in which KVM is used as a hypervisor [2]. There are different virtualization platforms are accessible in the present IT industry. Among such platforms some are open source while others are commercial hypervisors. In this paper KVM have been taken as open source

B. KVM / QEMU

KVM (Kernel Virtual Machine) is a Linux kernel module that allows a user space program to utilize the hardware virtualization features of various processors. KVM is a Type-2 hypervisor running on Linux, and it uses Linux kernel as a bare metal hypervisor. A host running KVM is actually running a Linux kernel and the KVM kernel module [3].

QEMU is an open source machine emulator which uses translation or virtualization to run operating systems or programs for various machine architectures. In addition to emulating CPU architectures, QEMU also emulates platform devices. QEMU combines a CPU and a set of [4]

Under kvm, virtual machines are created by opening a device node (/dev/kvm.) A guest has its own memory, separate from the user space process that created it. A virtual cpu is not scheduled on its own,

- */dev/kvm Device Node*

kvm is structured as a fairly typical Linux character device. It exposes a /dev/kvm device node which can be used by userspace to create and run virtual machines through a set of ioctl()s.

- *The operations provided by /dev/kvm include:*
  1. Creation of a new virtual machine.
  2. Allocation of memory to a virtual machine.
  3. Reading and writing virtual cpu registers.
  4. Injecting an interrupt into a virtual cpu.
  5. Running a virtual cpu.

Fig 2 shows how guest memory is arranged. Like user memory in Linux, the kernel allocates discontiguous pages to form the guest address space [5]

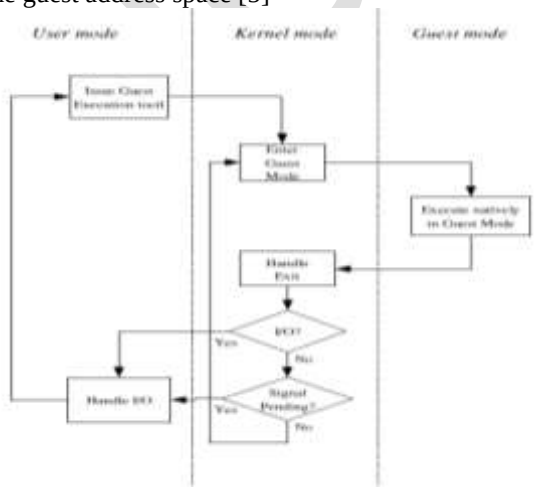

Fig 2: Guest Execution Loop

C. x86 Hardware Virtualization Support

Beginning in late 2005 when Intel started shipping IntelVT extensions, x86 processors gained the ability to have both a host mode and a guest mode. These two modes co-exist with the four levels of segment privileges and two levels of page privileges. AMD also provides processors with similar features with their AMD-V extensions. Intel-VT and AMD-V are similar in their major features and programming structure. With Intel-VT and AMD-V, hardware ring compression and binary translation are obviated by the new hardware instructions. The first generation of Intel and AMD virtualization support remedied many of the functional shortcomings of x86 hardware, but not the most important performance shortcomings. Second-generation virtualization support from each company addresses key performance issues by reducing unnecessary translation look aside buffer (TLB) flushes and cache misses and by providing multi-level memory management support directly in hardware. KVM provides a user-space interface for using AMDV and Intel-VT processor instructions. Interestingly, KVM does this as a loadable kernel module which turns Linux into a virtual machine monitor [4].

The AMD-V and Intel-VT instructions available in newer processors provide a mechanism to force the processor to trap on certain sensitive instructions even if the processor is running in privileged mode. The processor uses a special data area, known as the VMCB or VMCS on AMD and Intel respectively, to determine which instructions to trap and how the trap should be delivered. When a VMCB or VMCS is being used, the processor is considered to be in guest mode. KVM programs the VMCB or VMCS to deliver sensitive instruction traps back into host mode. KVM uses information provided in the VMCB and VMCS to determine why the guest trapped and emulates the instruction appropriately. Since this technique does not require any modifications to the guest operating system, it can be used to run any x86 operating that runs on a normal processor [4].

D. Libvirt

The libvirt is a virtualization abstraction layer, which is able to manage a set of virtual machines across different hypervisors. The goals of libvirt are to provide a library that offers all necessary operations for hypervisor management without implementing functionalities, which are tailored to specific virtualization solutions and which might not be of general interest. Additionally, the long-term stability of the libvirt API helps these management solutions to be isolated from changes of hypervisor APIs.

Libvirt is an abstraction layer library for various hypervisor management APIs written in C. It started as a Xen wrapper, but has been extended to several other hypervisors later on. It is conceptual divided into a hypervisor agnostic and several hypervisor specific parts, also called drivers.

Applications can use libvirts public API, which internally maps to appropriate driver functions through an internal driver API (see also Fig. 3). This driver model has proven to be very flexible and easy to extend, so drivers for storage and network interfaces using the same model have been added [6].

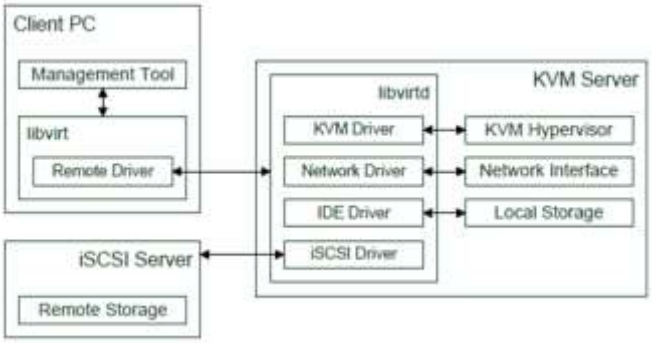

Fig 3: Driver based libvirt architecture

II. DEPLOYMENT

The deployment of the virtualization on Ubuntu 12.04 LTS using KVM hypervisor and Libvirt. All following steps with the command and description will lead towards successful implementation.

1. Update the Ubuntu operating system properly
   sudo apt-get update

2. Install kvm hypervisor

   kishor@kishor-Ideapad-Z570:~$ sudo apt-get install kvm

3. Testing of kvm installation
   kishor@kishor-Ideapad-Z570:~$ sudo kvm-ok

   Output will be:
   INFO: /dev/kvm exists
   KVM acceleration can be used

4. Install libvirt

   kishor@kishor-Ideapad-Z570:~$ sudo apt-get install libvirt-bin

5. (optional step) for adding user to libvirtd group.

   kishor@kishor-Ideapad-Z570:~$ sudo adduser $USER libvirtd

6. virt installing is actual installation of VM

   kishor@kishor-Ideapad-Z570:~$ sudo apt-get install virtinst

7. virt viewer installation

   kishor@kishor-Ideapad-Z570:~$ sudo apt-get install virt-viewer

8. installing qemu-kvm

   kishor@kishor-Ideapad-Z570:~$ sudo apt-get install qemu-kvm

9. copy os(linux/windows) iso file on home directory of ubuntu

10. virt installation with operating system iso

    kishor@kishor-Ideapad-Z570:~$
    sudo virt-install -n xp3 -r 512 –disk path=/var/lib/libvirt/images/xp3.img, bus=virtio, size=10 -c /home/kishor/xppro.ISO --accelerate --network network=default,model=virtio --connect=qemu:///system --vnc --noautoconsole –v

11. virt viewer for installing os on vm
    kishor@kishor-Ideapad-Z570:~$
    virt-viewer -c qemu:///system xp3

    Use the following commands if required

    1. delete the lock file with the following command
       sudo rm /var/lib/apt/lists/lock

    2. to delete the lock file in the cache directory
       sudo rm /var/cache/apt/archives/lock

    3. if any package is not updating or installing even internet conn is proper then use this command

       sudo sed -i '/^deb http:\/\/packages.medibuntu.org*/d' /etc/apt/sources.list /etc/apt/sources.list.d/*.list

    4. bridging: to allow virtual machines direct access to the outside network

       sudo apt-get install bridge-utils

    5. to install ssh certificates related task
       sudo apt-get install ssh

III. CONCLUSIONS

kvm is currently only implemented for the i386 and x86-64 architectures. However, other architectures such as powerpc and ia64 support virtualization, and kvm could be enhanced to support these architectures as well. kvm brings an easy-to-use, fully featured integrated virtualization solution for Linux. Its simplicity makes extending it fairly easy, while its integration into Ubuntu allows it to leverage the large Linux feature set and the tremendous pace at which Linux is evolving.

The learning aspirants can implement and study the various terminologies related to the cloud computing as well as virtualization.

ACKNOWLEDGMENT

We would like to thank the libvirt online portal for helpful information as well as Finolex Academy of Management and Technology, Ratnagiri and GJC Ratangiri for providing a generous hardware.

REFERENCES

[1]. Performance Comparison of Commercial VMM:ESXI, XEN, HYPER-V & KVM Computing for Sustainable Global Development (INDIACom)

[2]. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More, (2013).

[3]. Assessment of Hypervisor Vulnerabilities 2016 International Conference on Cloud Computing Research and Innovations 978-1-5090-3951-7/16 $31.00 © 2016 IEEE DOI 10.1109/ICCCRI.2016.19.

[4]. Using KVM to run Xen guests without Xen 2007 Linux Symposium, Volume One

[5]. kvm: the Linux Virtual Machine Monitor

[6]. Non-intrusive Virtualization Management using libvirt Matthias Bolte, Michael Sievers, Georg Birkenheuer, Oliver Niehörster and André Brinkmann Paderborn Center for Parallel Computing PC2,University of Paderborn, Fürstenallee 11, 33102 Paderborn, Germany {photon, msievers, birke,nieh, brinkman}@uni-paderborn.de 978-3-9810801-6-2/DATE10 © 2010 EDAA