

**Research Paper**

**Overview of Virtualization technologies and in-depth analysis of Intel Virtualization technology**

**Members:**

**Moazzan Sabir**

**Song Young Woong**

**Umer Abdul Khaliq**

**Ammar Khan**

**Submitted to : Sir Abdul Hannan**

**Abstract:**

Virtualization is a technology used to virtualize physical resources and provide an abstraction layer to running guest operating systems which provides a more logical way of pooling physical resources. We need virtualization because we cannot run multiple instances of operating systems on a single machine. In large scale organizations,we need multiple environments to develop/deploy many different software/services. Virtualization is cost effective and rapid deployments are made easy compared to direct methods and generation and revival of backups on-the-fly adds to its extensive lists of features. We can pool physical resources to each guest OS accordion to their need. There are different flavors of visualizations available. The software/thin-OS that provide these capabilities to a virtual machine is called a Hypervisor.

**Hypervisor:**

A hypervisor is a layer of abstraction between the hardware and the guest OS in a way that each guest sees a VM instead of the real hardware. It only contains a kernel to boot up along with an IP address (to setup its management console) and few MBs of ram to operate and it is packed with guest management tools to assist guest virtualization. To run multiple guests, a hypervisor is packed with a Hyper call layer for trap and emulation, I/O virtualize manager (could be packed into guest OS),Interrupts handler/router, a page mapper to points the hardware to pages for a particular guest OS and a scheduler to transfer control back and forth between hypervisor and guest OSes.

**A) Type-1 Hypervisors:**

Also called bare-metal hypervisors runs directly onto the physical hardware. It doesn’t needs an OS to run. It contains its own “Virtual Machine Monitor” (VMM) which sits between the physical hardware and guest OSes. Further,they are divided into two sub-categories named Monolithic and Micro-kernelized,but the key difference is only how they manage device drivers. Main focus of the paper is Type-1 along with Hybrid-Hypervisor.



**B) Type-2 Hypervisors:**

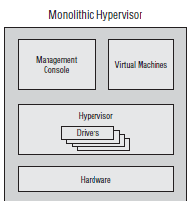
Also called hosted hypervisors as they run on a hosted operating system with a virtualization software running onto it. Main disadvantage is that it is not as effective as Type-1 in terms of scalability and robustness and management.



**Hypervisor Kernel Designs:**  
 There are two approaches to design a kernel for a hypervisor.

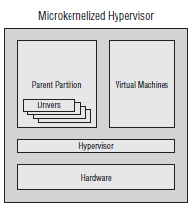
**A) Monolithic Approach:**

The way of handling device drivers is done by embedding all the drivers for the hardware within the hypervisor layer for all the virtual machines. It doesn't require a hosted OS or a micro-kernel to handle drivers which is its biggest advantage. The limitation is the hypervisor is limited to driver support and if one driver crashes,it is going to crash each VM using that specific driver.



**B) Micro-Kernelized Approach:**

The way of handling device drivers is done by embedding them in a micro-kernel of the hosted OS running in the parent partition. There is no need to install device drivers for each guest VM running in guest partition. Parent partition holds all the device drivers and it provides wide range of OS support and device drivers. Main disadvantage is that,if parent partition micro-kernel crashes,it upsets all the guest VMs. Parent partition contains device drivers and only exposes generic classes of virtual drivers to VMs and provides a transport medium to make the communication of guest to hardware possible.



**Virtualization Key Concepts:**

Before diving deep,lets take a look at the key concepts and strategies used to implement the different aspects of virtualization.

**A) Trap and Emulate:**

Virtual Machines are unprivileged-users created on a machine. This means they cant execute all the instructions. When binary is being passed on to the CPU,hypervisor traps the sensitive instruction,that a VM cant execute considering its privileges,and directs a new instruction call on its own level to cpu and returns the results. In this way,no binary modification has to be done in guest OS.

**B) Intel VT-x:**

Hardware assisted virtualization technique that detects a sensitive instruction from a Virtual Machine and changes its mode of execution from VMX non-root to VMX root mode. When VMX root mode is set in action,the hypervisors calls the instruction on the behalf of the VM thus enabling the execution of sensitive instructions.

**C) VT-d:**

It provides I/O device assignment. This feature allows an administrator to assign I/O devices to VMs in any desired configuration directly, MA remapping which supports address translations for device DMA data transfers, Interrupt remapping which provides VM routing and isolation of device interrupts and reliability features which reports and records system software DMA and interrupt errors that may otherwise corrupt memory of impact VM isolation.

**D) Hypercall:**

A hyper call is a software trap from a domain (guest OS) to the hypervisor.

**Ring Challenges:**

X86\_X64 architecture offers 4 levels of privileges named

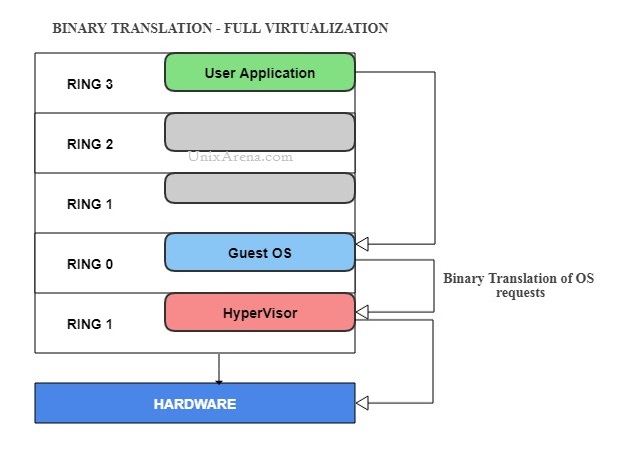
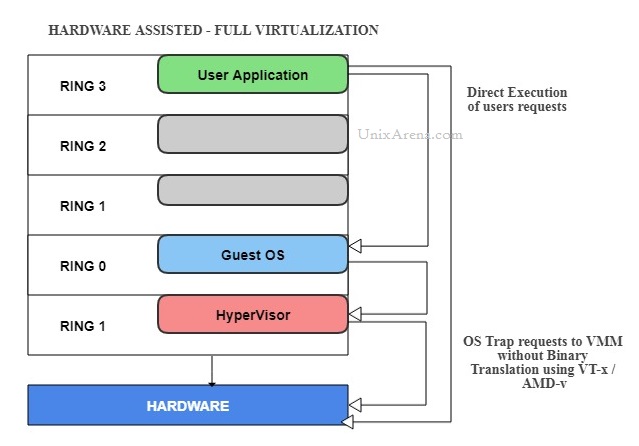
1. Ring 0
2. Ring 1
3. Ring 2
4. Ring 3

The OS itself sits on the Ring 0 hence having most privileges and having full control of the hardware and the applications run on Ring 3 having less privileges. In Virtualization,VMM is placed in Ring 0 and and VM is placed in Ring 1.Ring 3 contains user applications. Three main resources being protected: memory, I/O ports, and the ability to execute certain machine instructions.

About 15 machine instructions, out of dozens, are restricted by the CPU to ring zero. Many others have limitations on their operands. These instructions can subvert the protection mechanism or otherwise foment chaos if allowed in user mode.

**Modes of Virtualization:**

There are two modes of virtualization named

****

**Full Virtualization:**

In this mode of virtualization,each individual,isolated guest OS have no clue of being run in a virtual mode. The hypervisor layer provides total abstraction of hardware to the guest OS as if it is running on an isolated bare-metal. It is further subdivided into two categories.

**A) Software Assisted Full Virtualization:**

The VMM traps the sensitive instructions from guest OS and use its privileges to carry out the working via “Trap and Emulate” procedure.

**B) Hardware Assisted Full Virtualization:**

Employing VT-x architecture of CPU,guest OS can be trapped by the CPU itself and elevate the mode of execution.

**Ring -1 ???:**

As you can see in HW-Assisted mode, the VMM is launched below ring 0. INTEL added another ring below the ring 0 to host the Virtual Machine Monitor to set up some control structures (called VMCS) defining the virtual machines and then call “vmenter” and handle “vmexit” i.e. conditions on which the virtual machine needs help from the hypervisor.

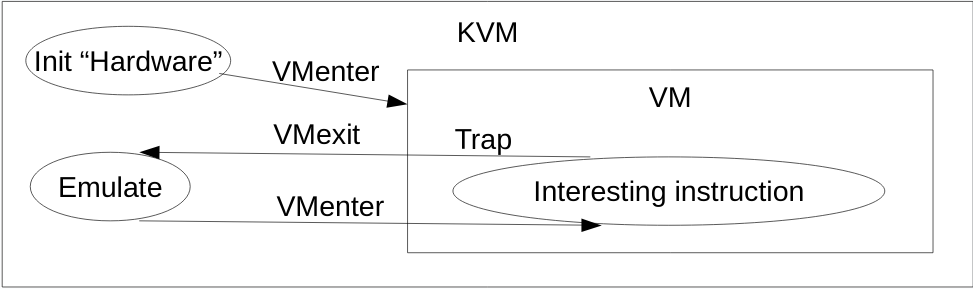
This piece of code is referred to as "ring -1". There is no such actual privilege level, but since it can host multiple kernels all of which believe they have ring 0 access to the system

Examples of emulated instructions:

–Access to “hardware device” (MMIO / PIO)

–Special instructions (CPUID, LGDT)

–Access to interesting “registers” that have side effects



**Intel VT FlexPriority:**

VT FlexPriority is a processor extension that optimizes virtualization software efficiency by improving interrupt handling. With virtual processor IDs (VPID), a VM IDtag in the CPU hardware structures associates cache lines with each VM actively running on the CPU.

When passing through “vmenter” to “vmexit” phases, the processor load/flush caches associated with particular VMs.

This permits the CPU to flush only the cache lines associated with a particular VM when itis flushed from the CPU, avoiding the need to reload cache lines for a VM that was not migrated and resulting in lower overhead.

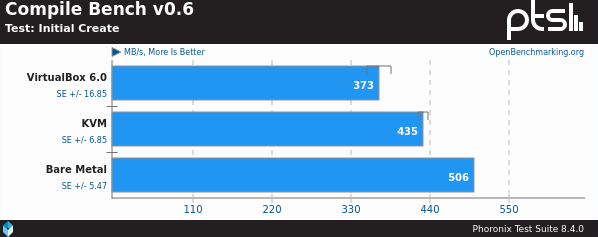
**Market Virtualization solutions Comparison:**

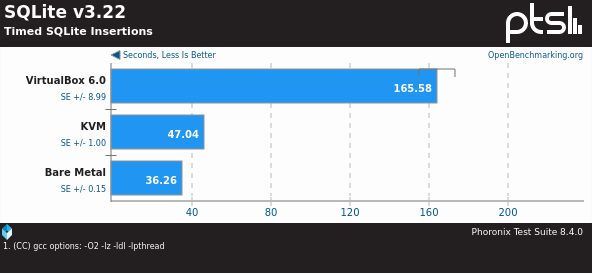
a) Kernel Virtual Machines ( HW-Assisted )

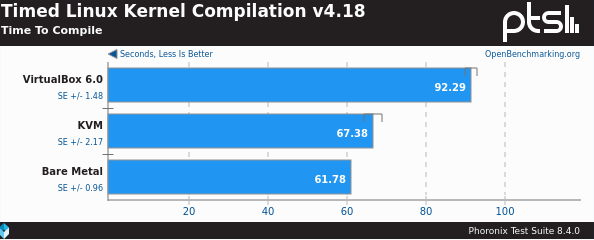
b) VirtualBox ( Binary Trap )

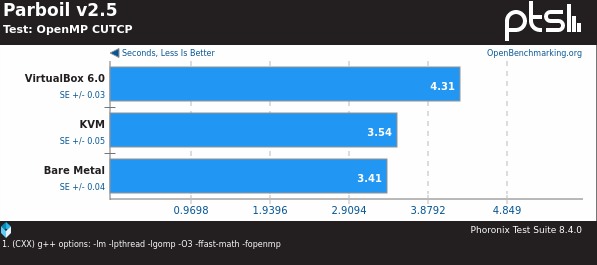
c) Bare Metal ( Type 1 )

**Test Results:**









**Why HW-Assisted Virtualization?**

1)When any of the 17 instructions controlling critical platform resource arises, but the OS is not running in Ring 0, this could cause conflict resulting in system fault of wrong response.

2) Run time modification forces the VMM to provide complex workarounds during operations, which can impact performance and system reliability.

3) Para-virtualization prevents VMM from hosting unmodified guest operating system

4) Both run time modification and para-virtualization require extensive software modification efforts from the VMM and the OS vendors. This increases the cost and complexity of IT support.

**Website References:**

><https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>

><https://www.unixarena.com/2017/12/para-virtualization-full-virtualization-hardware-assisted-virtualization.html/>

> <https://www.ibm.com/developerworks/library/l-hypervisor/index.html>

><https://twiki.cern.ch/twiki/bin/view/Virtualization/KVM>

><https://virtualizationreview.com/Blogs/Mental-Ward/2009/02/KVM-BareMetal-Hypervisor.aspx>

> <http://www-archive.xenproject.org/files/Marketing/HowDoesXenWork.pdf>

><https://www.vmware.com/co>

https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware\_paravirtualization.pdf

>https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/virtualization-enabling-intel-virtualization-technology-features-and-benefits-paper.pdf

[ntent/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware\_paravirtualization.pdf](https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf)

> https://developer.ibm.com/articles/cl-hypervisorcompare/

**References of Research Articles:**

> VIRTUALIZATION IN CLOUD COMPUTING

<https://page-one.springer.com/pdf/preview/10.1007/978-3-642-29737-3_25>

>Kernel-Based Virtualization Technology <https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol47-3/paper18.pdf>

> An Evaluation of KVM for Use in Cloud Computing <https://people.cs.clemson.edu/~jmarty/papers/icvci08.pdf>

> Performance Issues in Cloud Computing: KVM Hypervisor’s Cache Modes Evaluation <https://www.uni-obuda.hu/journal/Dordevic_Macek_Timcenko_60.pdf>

# > System Performance Evaluation of Para Virtualization, Container Virtualization, and Full Virtualization Using Xen, OpenVZ, and XenServer <https://ieeexplore.ieee.org/document/6906035>

# > Virtualization Maturity Reference Model for Green Software

# <https://ieeexplore.ieee.org/document/6413772>

# > Overview of virtualization in cloud computing

# <https://ieeexplore.ieee.org/document/7570950>

# > Virtualization in Enterprises' Different Growth Stages and Its Motives: A Classical Grounded Theory Research

# <https://ieeexplore.ieee.org/document/6923674>

# >http://dsc.soic.indiana.edu/publications/virtualization.pdf