

**Research Paper**

**Overview of Industrial Virtualization Technologies and Evolution of Hybrid-Hypervisor**

**Ammar Khan 17010020010**

**Department of Computer Science**

**UMT-Sialkot**

**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 opearting systems on a single machine.In large scale organizations,we need multiple environments to develp/deploy many different softwares/services.Virtualization is costeffective 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 accoriding to their need.There are different falvours of vritualizations 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.

**Type-1 Hypervisors:**

Also called bare-metal hypervisors runs dirrectly 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-catagories 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.



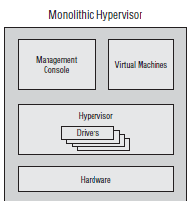
**Type-2 Hypervisors:**

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



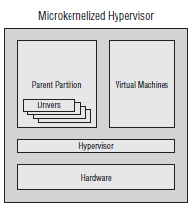
**Monolithic Approach:**

The way of handeling device drivers is done by embedding all the drivers for the hardware within the hypervisor layer for all the virtual machines.It doesnot require a hosted OS or a micro-kernel to handle drivers which is its biggest advantage.The limitaion is the hypervisor is limited to driver support and if one driver crashes,it is going to crash each VM using that specific driver.



**Micro-kernelized Approach:**

The way of handeling 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 supprot 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 trasport 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 virtualizations.

**Trap and Emulate:**

Virtal Machines are unprivellaged-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 sensative instruction,that a VM cant execute considering its privillages,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.

**Intel VT-x:**

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

**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.

**Hypercall:**

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

**Ring Challenges:**

X86\_X64 architerchter offers 4 levels of privellages named

>Ring 0

>Ring 1

>Ring 2

>Ring 3

The OS itself sits on the Ring 0 hence having most privellages and having full control of the hardware and the applications run on Ring 3 having less privellages.In Virtualization,VMM is placed in Ring 0 and and VM is placed in Ring 1.Ring 3 contains user applications.

**Modes of Virtualization:**

There are two modes of virtualizations named

**>Full Virtualization**

**>Para Virtualization**



**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 abstration of hardware to the guest OS as if it is running on an isolated bare-metal.It is further subdivded into two catagories.

**>Software Assisted Full Virtualization:**

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

**>Hardware Assisted Full Virtualization:**

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

**Para Virtualization:**

The working of this mode of virtualization is a lot different from FV technique.Guest OSes knows that they are being virtualized and they are modifed for it.They don’t rely on binary traps.To para virtualize,it requires a virtualization layer (hypervisor) along with a host OS called Domain 0 (Dom0) , Driver Domain.All other guest OSes that are virtualizaed on child partition are called DomainU (DomU).

**>Domain0 (Dom0):**

An operating system that sits on parent partition which runs in Ring 0 but sits above the hypervisor layer.It is the first OS that is booted and it controls and manages all other DomU.Dom0 has direct physical access to all hardware and exports generic class drivers to DomU for communication.Hypercalls(syscalls in regular OS) are used for the communication of sensitive instructions between Dom0 and and guest OSes

**>Driver Domain:**

It is a guest OS which belongs to privellaged class and is responsible for exposing and commuting driviers to all other guest OSes.It is responsible of sharing of all the physical of devices among other guest OSes.It is loaded as a separate part because driver communications is the most error-prone aspect of a system.So,if driver domain crashes,Dom0 remains unaffected and it can relaunch or recover driver domain.

**>Domain guest (DomU):**

All other virtualized OSes on child partition are called DomU.These are unprivellaged OSes.The OSes running on child partition knows they are being virtualized and they are modified for it.

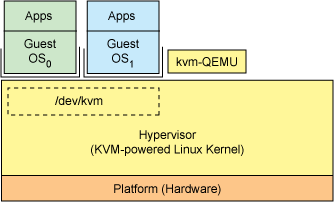
**Hybrid-Hypervisor:**

Since,we have covered all the mediums to create a virtual enviroment.Hybrid-Hypervisor channelized all the technologies alltoggether to cover every aspect of virtualization.Linux implemented brought these capability into their kernel as modules.Since,linux is an open-source OS,so code for para-virtualization can be modified in the kernel-module and speed-based code can be implemented as required.On the other hand,it takes full advantage of hardware-assisted virtualization via another component.

**Implementation on Linux:**

When KVM module is loaded, it transforms the linux kernel into a hypervisor layer providing hardware-assisted virtualization via VT-x and VT-d.Linux OS itself monitors and administers the guest oses and these VMs are just another process to linux.Linux exposes its /dev/proc hardware files to guest OS and provided direct access to them.It manages “trap and emulate” via its own linux kernel because the VMs are like processes running natively on its own.KVM provides user space to these VMs and hence provide both HW and Soft assisted virtualization methods.

The other feature of KVM is that it is packed with QEMU(modified version). For paravirtualizaton,linux kernel on the instruction of KVM module creates a parent partition where QEMU OS resides(Dom0) and other guest OSes are created on child partitions and QEMU(Dom0) acts a coordinator between linux kernel(hypervisor-layer) and the guest OSes (DomU) and implemets Hypercalls thus providing emulation.



**Key Benefits of Hybrid Hypervisor:**

Key benefit of this type of hypervisor is that it is managed by linux kernel.Linux is long known for its large support for device drivers,optimized kernel code,efficient management of processes/memory and high speed scheduler to switch I/O and it is optimized for much larger hardwares and softwares solutions.It provides all types of virtualizations with easy implementation and managibility.They key adanvtage of this implemetation is that each VM created is a process to linux.So,it is a lot easy to manage and control a VM plus it is open-source ,so you can adjust each aspect and optimize it according to your need.

**Conclusion:**

In a world where OSes are reduced to embeded,containers and dockers ,where the OSes are just packed to support a single application and its role is only to provide the required environment,one cannot keep switching between the hypervisors for changing demands of different OS virtualiaztion support.Hybrid Hyperviors can be a relief for a cloud architect to manage lots of changing scenarios ,changing techniques and changing deployments criteria by providing multi-dimention implementations of virtualization and it could relief a system administrator by providing linux process-like managemnet of VMs.Regardless of relying on closed source and limited support solutions,Linux KVMs provide both multi-scale and easily managable solution virtualization.

**References of Websites:**

><https://searchservervirtualization.techtarget.com/definition/paravirtualization>

><https://blogs.technet.microsoft.com/chenley/2011/02/23/hyper-v-microkernelized-or-monolithic>

><https://wiki.xenproject.org/wiki/Hypercall>

><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/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