CPU Virtualization and its Techniques

Rajbeer Kaur[1], Asst. Prof Er. Harkirat Singh Brar[2]

Department of Computer Science Engineering, MIMIT MALOUT

---***---

Abstract -Virtualization provides the crowded data centres and underutilized servers the power to combine multiple physical servers into a single server that is able to run multiple virtual machines and enables the physical server to run at a higher rate of utilization. Virtualization is the process in which a virtual version of something (hardware platforms, storage, network resources) is created. In this review paper we made an effort to study about CPU virtualization and its techniques. In CPU virtualization, a single processor acts as multiple processors. It can be performed in two ways: paravirtualization and full virtualization. In paravirtualization changes are made in the Operating System. On the other hand, in full virtualization the Operating system remains the same. On comparing these two it is observed that paravirtualization is less portable than full virtualization but the overhead in paravirtualization is much lesser than full virtualization which makes it a better solution for CPU virtualization.

***Key Words*:**Virtualization, CPU virtualization, Virtualization techniques

1.INTRODUCTION

Virtualization is a process of creating a virtual version of a server, desktop, operating system, networking, applications or a storage device.Virtualization uses software to create an abstraction layer over computer hardware that allows hardware elements of single computer(processor, memory, storage and more) to be divided into multiple virtual computers, called virtual machines(VMs). Each VM works like an independent computer and is able to run its own operating system even though it is working on a small segment of actual computer hardware. Hypervisor is a software that makes virtualization feasible. Hypervisor runs on a physical server or host. VMs can be built using hypervisor. VM is a software based machine which provides facilities of a physical server.

VMs have operating system, applications and are completely independent of one another. Hypervisor helps in management of the resources allocated to multiple VMs from the physical server. Different operating systems can run on different VMs as these are independent. VMs can be moved from one hypervisor to another one on a different server which makes them extremely portable. A hypervisor installs virtualization layer on the system with direct access to hardware resources without touching an operating system. This makes hypervisor more productive than a hosted architecture. The functionality of hypervisor varies with design and execution. Each Virtual Machine Monitor(VMM), that runs on hypervisor applies the VM hardware abstraction andtakes the responsibility of running a guest OS. VMM partitions the CPU, memory and I/O devices in order to virtualize the system [2].

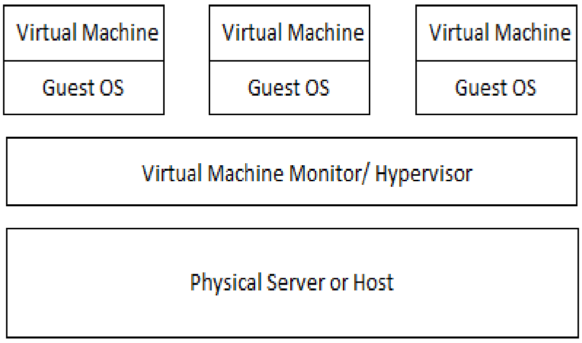


Fig -1: Core of Virtualization

2. CPU VIRTUALIZATION

CPU Virtualization enables a single processor to act like it has more than one CPUs. This is helpful to utilize the CPU power in a more efficient manner which makes the CPU run faster. CPU Virtualization is the main requirement for virtual machine software [3]. X86 operating systems suppose that they have full control on computer hardware as they can run directly on bare metal hardware. X86 architecture provides four levels of privilege known as Ring 0, 1, 2 and 3. Higher the ring number, lower the privilege of instruction being executed. The user level applications execute in ring 3. Ring 1 and 2 are kept as null for future purpose. OS is placed in Ring 0 so that it can directly access the storage and hardware and can manage the hardware and privileged instructions. A virtualization layer should be placed under OS to virtualize x86 architecture which helps to create and manage the VMs. It is difficult to virtualize sensitive instructions. This difficulty in trapping and translating these sensitive instructions at runtime made virtualization more difficult. [VMware article]
Alternative techniques are developed for handling this kind of instructions. Full virtualization and paravirtualization are few of them.

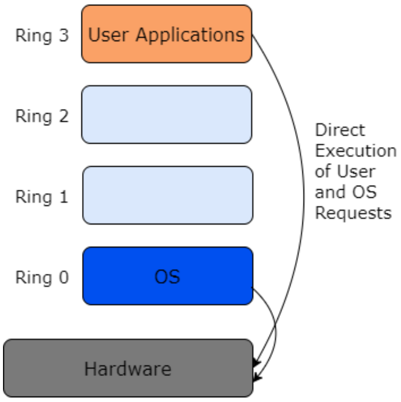


Fig -2: X86 architecture without Virtualization

2.1. Full Virtualization

In this host OS runs directly on hardware and the guest OS runs on VM. Thus, guest OS is completely separated from the underlying hardware. Here guest OS is unaware of the fact of being virtualized and remains unmodified.

Binary translation is used in full virtualization. A hypervisor is installed in ring 0 which manages underlying hardware.

Guest OS runs in layer 1. When sensitive instructions are called, hypervisor will use binary translation to stop them.

Full virtualization is useful for sharing a server with multiple users and separating users from each other. It provides security and isolation to VMs. Similar Guest OS can run virtualized which makes it portable. Products that apply this technique are VMware's virtualization products and Microsoft Virtual Server.

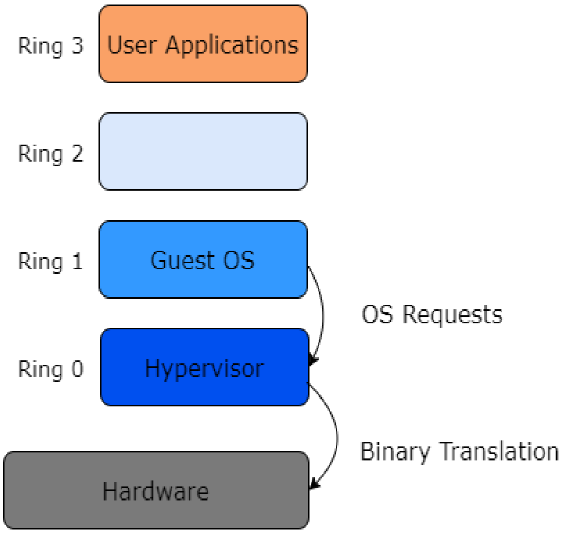


Fig -3: Full Virtualization Process

2.2. Paravirtualization

In paravirtualization, guest OS knows that it is guest and is available with another OS. Here guest OS are paravirtualized and an intelligent compiler assists them, so that non-virtualizable instructions can be replaced by hypercells [4]. Guest operating system is attached with special software known as drivers which plays an important role in communication with hypervisor. Drivers are installed on guest OS and passes the guest OS commands to hypervisor and hypervisor forwards these commands to the host operating system. The main purpose of paravirtualization is to reduce overhead and improve system's performance.

In this guest OS run in ring 0. Here some modifications are made in OS kernel which is the major disadvantage of this technique. The call for sensitive instructions is passed to hypervisor and this call is known as hypercell. Sensitive instructions called by guest OS are turned into hypercalls but hypervisor still supervises the system hardware resources.
Xen/Xenserver is the most popular hypervisor used for paravirtualization [4].

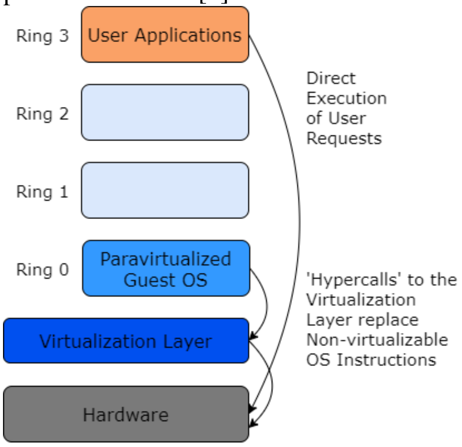

Ring 3
User Applications
Ring 2
Ring 1
Ring 0
Paravirtualized Guest OS
Virtualization Layer
Hardware
Direct Execution of User Requests
'Hypercalls' to the Virtualization Layer replace Non-virtualizable OS Instructions

Fig -4: Paravirtualization Process

3. FULL VIRTUALIZATION VSPARAVIRTUALIZATION

Operating system remains untouched in full virtualization whereas some modifications are made in operating system in order to do paravirtualization. Full virtualization faces low performance during binary translation. Therefore, products like Xen, KVM employ paravirtualization.
In Full virtualization, processing power is reserved in the physical server as hypervisors have their own processing needs. Server also needs to reserve resources to run hypervisor applications. This affects server performance and slows down applications. On the other hand, in paravirtualization, less processing power is required by hypervisor to manage guest OSes as guest OS already knows the needs of OSes placed on physical server. Thus, overhead in paravirtualization is lesser as compared to full virtualization[New]. Paravirtualization faces other problems like less compatibility and portability as there must be any support for unmodified OS. Only Linux, FreeBSD can be virtualized. Maintenance cost of paravirtualized OSes is very high as OS kernel modification is required. Also, there is a change in paravirtualization's performance as the workload varies. Paravirtualization is an easy and more practical solution as compared to full virtualization.[5]

3.1. Comparison of Paravirtualization and full virtualization using Xen

Xen is an open source bare metal hypervisor used to run multiple instances of an OS or different OSes parallelly on a single server. Xen is used in many commercial and open source applications.
Xen environment consists of several parts. Some basic parts are Xen hypervisor, the Dom0, VM guests and tools, commands and configuration files for virtualization management.
The Xen hypervisor is an open source software used to maintain coordination of low-level interaction between VMs and hardware. Dom0 is the host environment,

that provides a special domain to provide management environment. Xen based VM is also known as DomU. Managements tools, commands and configuration files are a combination of GUI tools and commands.

3.1.1. Experimental Setup

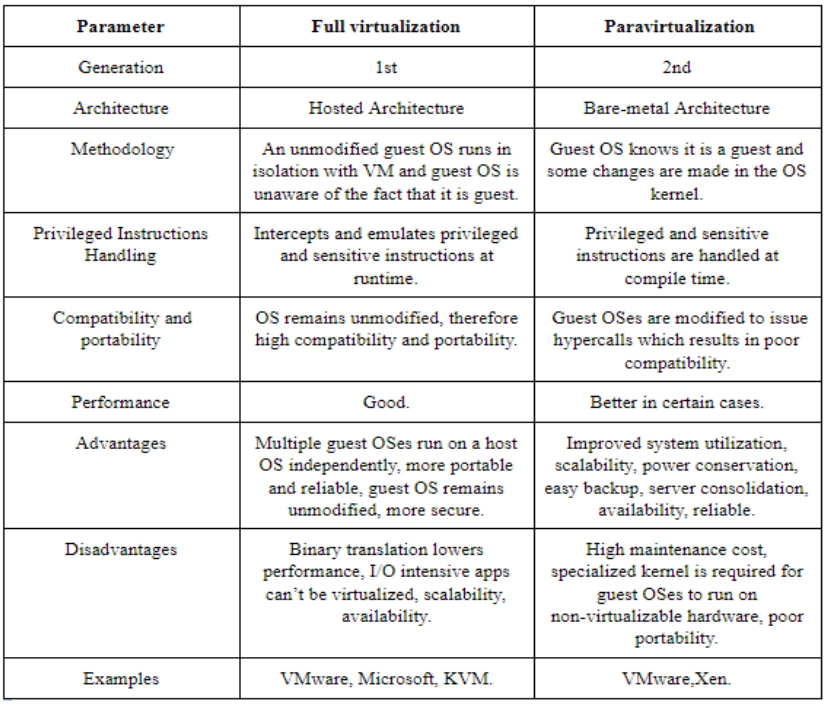
For the evaluation process, Xen 4.2.1 is used. Dom0 is running an open source OS of version 12.3. Guest OS running in full virtualization and paravirtualization VMs is Linux PREEMPT-RT which is open source and configurable.

Tests are performed on each paravirtualized and full virtualized VMs separately. Tested paravirtualized VM is known as Under Test Para-Virtualized Machine (UTPVM) whereas a tested fully virtualized VM is known as Under Test Fully-Virtualized Machine (UTFVM).

3.1.2. Testing Results

Clock tick duration examines clock tick processing duration in kernel. A real time highest priority thread is created. This thread performs a finite loop of tasks like get time using RDTSC instruction, to start a busy loop, get time again and again using same instruction. The overhead is increased by 310% for UTFVM and 270% for UTPVM.
Results show that Xen default setting increases the worst-case latency by 74%. The system bus load impact is larger in full virtualization as compared to paravirtualization. The overhead in paravirtualization is 35% lesser than full virtualization [6].

Table -1: Full Virtualization Vs Paravirtualization



Parameter	Full virtualization	Paravirtualization
Generation	1st	2nd
Architecture	Hosted Architecture	Bare-metal Architecture
Methodology	An unmodified guest OS runs in isolation with VM and guest OS is unaware of the fact that it is guest.	Guest OS knows it is a guest and some changes are made in the OS kernel.
Privileged Instructions Handling	Intercepts and emulates privileged and sensitive instructions at runtime.	Privileged and sensitive instructions are handled at compile time.
Compatibility and portability	OS remains unmodified, therefore high compatibility and portability.	Guest OSes are modified to issue hypercalls which results in poor compatibility.
Performance	Good.	Better in certain cases.
Advantages	Multiple guest OSes run on a host OS independently, more portable and reliable, guest OS remains unmodified, more secure.	Improved system utilization, scalability, power conservation, easy backup, server consolidation, availability, reliable.
Disadvantages	Binary translation lowers performance, I/O intensive apps can't be virtualized, scalability, availability.	High maintenance cost, specialized kernel is required for guest OSes to run on non-virtualizable hardware, poor portability.
Examples	VMware, Microsoft, KVM.	VMware,Xen.

4. CONCLUSION

Virtualization is a process that enables to install multiple OSes on a hardware. In this review paper, we discussed CPU virtualization and its techniques paravirtualization and full virtualization. When these two are compared it is observed that overhead in paravirtualization is much lesser than full virtualization which makes it an easy and practical solution. Xen experimentation also proved that system bus load and overhead is much lesser in paravirtualization.

REFERENCES

1.IBM Cloud Education, 19 June 2019,What isVirtualization?IBM. 3 Feb 2020,https://www.ibm.com/cloud/learn/virtualization-a-complete-guide?mhsrc=ibmsearch_a&mhq=virtualization.
2. Matthew Portnoy (2012). Virtualization Essentials. North Carolina. John Wiley & Sons. P 1-2.
3. Lawrence Abrams, 28 Nov 2017, How to Enable CPU Virtualization in Your Computer's BIOS. 15 Feb 2020, https://www.bleepingcomputer.com/tutorials/how-to-enable-cpu-virtualization-in-your-computer-bios/
4. VMWare, 11 Mar 2008, Understanding Full Virtualization, Paravirtualization and Hardware Assist. 20 Feb 2020, https://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html
5. S. Suresh, Dr. M. Kannan. A Study on System Virtualization Techniques. International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014). ISSN: 2347-8446, Volume 2, Issue 1 Jan- March 2014.
6. Hassan Fayyad-Kazan, Luc Perneel, Martin Timmerman. Full and Para-Virtualization with Xen: A Performance comparison. Journal of Emerging Trends in Computing and Information Sciences. ISSN: 2079-8407, Volume 4, Issue 9 Sep 2013.