

Rochester Institute of Technology

RIT Scholar Works

Theses

5-2014

Development of a virtualization systems architecture course for the information sciences and technologies department at the Rochester Institute of Technology (RIT)

Pooriya Aghaalitari

Follow this and additional works at: <https://scholarworks.rit.edu/theses>

Recommended Citation

Aghaalitari, Pooriya, "Development of a virtualization systems architecture course for the information sciences and technologies department at the Rochester Institute of Technology (RIT)" (2014). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

**DEVELOPMENT OF A VIRTUALIZATION SYSTEMS ARCHITECTURE COURSE
FOR THE INFORMATION SCIENCES AND TECHNOLOGIES DEPARTMENT AT
THE ROCHESTER INSTITUTE OF TECHNOLOGY (RIT)**

By

Pooriya Aghaalitari

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Networking and System Administration
at
The Rochester Institute of Technology (RIT)

B. THOMAS GOLISANO

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

May 2014

**DEVELOPMENT OF A VIRTUALIZATION SYSTEMS ARCHITECTURE COURSE
FOR THE INFORMATION SCIENCES AND TECHNOLOGIES DEPARTMENT AT
THE ROCHESTER INSTITUTE OF TECHNOLOGY (RIT)**

By

Pooriya Aghaalitari

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Networking and System Administration
at

The Rochester Institute of Technology (RIT)

B. THOMAS GOLISANO

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

May 2014

Copyright by Pooriya Aghaalitari, 2014

Project Committee Approval

Name	Signature	Date
Charles Border		
Chair		
Tom Oh		
Committee Member		
Sharon Mason		
Committee Member		

Abstract

DEVELOPMENT OF A VIRTUALIZATION SYSTEMS ARCHITECTURE COURSE FOR THE INFORMATION SCIENCES AND TECHNOLOGIES DEPARTMENT AT THE ROCHESTER INSTITUTE OF TECHNOLOGY (RIT)

By Pooriya Aghaalitari, MSc

Rochester Institute of Technology (RIT), 2014

Virtualization is a revolutionary technology that has changed the way computing is performed in data centers. By converting traditionally siloed computing assets to shared pools of resources, virtualization provides a considerable number of advantages such as more efficient use of physical server resources, more efficient use of datacenter space, reduced energy consumption, simplified system administration, simplified backup and disaster recovery, and a host of other advantages. Due to the considerable number of advantages, companies and organizations of various sizes have either migrated their workloads to virtualized environments or are considering virtualization of their workloads. As per Gartner “Magic Quadrant for x86 Server Virtualization Infrastructure 2013”, roughly two-third of x86 server workloads are virtualized [1]. The need for virtualization solutions by companies and organizations has increased the demand for qualified virtualization professionals for planning, designing, implementing, and maintaining virtualized infrastructure of different scales. Although universities are the main source for educating IT professionals, the field of information technology is so dynamic and changing so rapidly that not all universities can keep pace with the change. As a result, providing the latest technology that is being used in the information technology industry in the

curriculums of universities is a big advantage for information technology universities. Taking into consideration the trend toward virtualization in computing environments and the great demand for virtualization professionals in the industry, the faculty of Information Sciences and Technologies department at RIT decided to prepare a graduate course in the master's program in Networking and System Administration entitled "Virtualization Systems Architecture", which better prepares students to find a career in the field of enterprise computing.

This research is composed of five chapters. It starts by briefly going through the history of computer virtualization and exploring when and why it came into existence and how it evolved. The second chapter of the research goes through the challenges in virtualization of the x86 platform architecture and the solutions used to overcome the challenges. In the third chapter, various types of hypervisors are discussed and the advantages and disadvantages of each one are discussed. In the fourth chapter, the architecture and features of the two leading virtualization solutions are explored. Then in the final chapter, the research goes through the contents of the "Virtualization Systems Architecture" course.

Dedication

I dedicate this thesis to my mother, Farzaneh, for her unconditional support, morale, and encouragement throughout my life. I also dedicate my thesis to my wife, Maryam, who stayed by my side and was patient throughout my graduate program.

Acknowledgements

I would like to highly appreciate my advisor, Dr. Charles Border, for his useful and continuous guidance throughout the graduate program. I would like to thank Dr. Tom Oh and Dr. Sharon Mason to be part of my committee. I also would like to thank Michelle Vaz and Ellen Yang who have always been available for assistance. Finally, I sincerely thank my family for their precious support and companionship.

Table of Contents

Introduction	9
Literature Review	12
Chapter 1 - Introduction to Virtualization	15
1.1. History of Virtualization	15
1.2. Distributed Computing Era.....	17
1.3. Moore's Law	19
1.4. Challenges of Distributed Environments	20
1.5. Virtualization Terminology	23
1.6. Virtualization Requirements.....	24
Chapter 2 - Challenges in the Virtualization of the x86 Platform.....	26
2.1. Ring Deprivileging.....	28
2.2. Virtualization Techniques.....	30
2.2.1. Full Virtualization.....	30
2.2.1.1. Full Virtualization Characteristics.....	32
2.2.2. Paravirtualization.....	32
2.2.2.1. Paravirtualization Characteristics.....	34
2.2.3. Hardware Assisted Virtualization	34
2.2.3.1. Hardware Assisted Virtualization Characteristics.....	36
Chapter 3 - Types of Hypervisors	37
3.1. Type 1 (Bare metal).....	37
3.1.1. Type 1 Hypervisor's Characteristics	37
3.2. Type 2 (Hosted)	39
3.2.1. Type 2 Hypervisor's Characteristics	40
3.3. Hardware Virtualization Extensions.....	41
3.3.1. Memory Management Unit (MMU) Virtualization.....	41
3.3.2. I/O MMU virtualization	44
3.3.3. Single Root I/O Virtualization.....	46
Chapter 4 - Virtualization Solution Leaders	51
4.1. VMware vSphere ESXi	53
4.1.1. Memory Management in VMware ESXi.....	55
4.1.1.1. Transparent Page Sharing (TPS).....	56
4.1.1.2. Ballooning.....	57
4.1.1.3. Memory Compression	59
4.1.1.4. Hypervisor Swapping.....	59

4.2. Microsoft Hyper-V	60
4.2.1. Memory Management	62
Chapter 5 - "Virtualization Systems Architecture"	65
5.1. Lecture1-Introduction to Virtualization.....	65
5.2. Lecture2-Introduction to VMware vSphere Infrastructure	66
5.3. Lecture3-Plan, Design, and Install vCenter Server	67
5.4. Lecture4-Designing, Creating, and Managing VMs in VMware vSphere	68
5.5. Lecture5-Networking in the vSphere Environment	68
5.6. Lecture6-Understanding High Availability in VMware vSphere.....	69
5.7. Lecture7-Introduction to Hyper-V.....	70
5.8. Lecture8-Designing, Creating, and Managing VMs in Hyper-V.....	70
5.9. Lecture9-Networking in Hyper-V	71
5.10. Lecture10-Understanding Hyper-V Replica	71
5.11. Lecture11-Building a Hyper-V Failover Cluster.....	72
5.12. Hands-on Labs.....	73
Conclusion	74
Appendixes	76
Appendix A – Virtualization Systems Architecture.....	76
Lectures of the Course	76
Hands-on Labs of the Course	76
Appendix B – Course Proposal Form.....	78
Appendix C – Course Syllabus	82
Bibliography.....	88
Tables of Figures	92

Introduction

As one of the hottest and most efficient technologies, virtualization has transformed the way computing is done in data centers. Old, siloed, and distributed data centers that were highly underutilized are now being utilized at a much higher rate using virtualization technology. As Daniels states in the article “Server Virtualization Architecture and Implementation”, Windows servers on average utilize 8 to 12 percent of physical server’s capacity and UNIX servers utilize 25 to 30 percent of physical server’s resources [2]. Virtualization allows companies to consolidate their workloads onto a smaller number of physical servers by deploying their workloads using Virtual Machines. Taking into consideration the smaller number of physical servers required to process the workloads, virtualization assists businesses in more efficiently using their data center space and reduce the cost associated with electricity and cooling. At the same time, since virtualization enables easy, fast, and on-demand server provisioning, it has aided in the management of test and development environments for various purposes and scales. Legacy systems do not need to run on legacy platforms any more as virtualization provides access to general-purpose hardware. As virtual machines are hardware agnostic, physical server upgrades can happen in hours if not in minutes. Finally, through high availability techniques provided by virtualization vendors, hardware failures are less noticeable by end users and can be responded to in a timely fashion.

Taking into consideration the cost-savings and advantages virtualization technology has introduced in the field of information technology, virtualization has turned into the kind of technology that quickly gains acceptance in

datacenter deployments. According to Gartner group, “the adoption rate of server virtualization in 2012 is predicted to be 14.3% of total new physical x86 servers and will reach 21.3% of total servers in 2016. Total virtual OS instances will contribute 70.2% of total OS instances in 2012 and reach 82.4% of total OSs in 2016” [3]. As dependence on virtualization has been growing, the demand for virtualization professionals in IT industry has been growing and virtualization has turned into a hot field. In order to provide students with world-class quality of education and prepare them to better suit the job market in the IT industry, the faculty of Information Sciences and Technologies at RIT have decided to create a course, which provides students with the core knowledge to plan, design, and implement a virtualized computing environment using industry standard virtualization solutions.

The course can be divided into three sections. The first section discusses about virtualization as a general technology including what virtualization is, when and why it came into existence, virtualization techniques and their differences, why a hypervisor is required to run multiple instances of an operating system on the same hardware, and advantages and disadvantages of virtualization. Although we would like not to be biased toward any vendor-specific solutions, that can't be done due the fact that the virtualization industry is led by a few giant vendors including VMware, Microsoft, and Citrix. In fact, as depicted in figure 1, VMware and Microsoft were the only two virtualization leaders in 2013 [4]. Since the beginning of the virtualization of the x86 platform, VMware has been the leading vendor in the virtualization industry. The second part of the course teaches students about VMware's main virtualization solution known as “VMware vSphere” and a number of its features. The final part of the course teaches

students about “Microsoft Hyper-V” and a number of its features. To assist the students in better understanding and learning the material as part of the second and third part of the course, all lectures come with complementary hands-on lab(s).

Figure 1. Magic Quadrant for x86 Server Virtualization Infrastructure



Literature Review

Although virtualization is an old concept and technology, it is just in the past few years that it has become an integral part of computing environments in many companies and organizations. As the widespread use of virtualization technology is relatively new in the IT industry, virtualization courses can't be found as part of the curriculums of many universities. At the same time, as the need for virtualization professionals is highly felt in the job market, there have emerged a number of virtualization courses offered by professional training institutes. What is similar among all the virtualization courses offered by professional training institutes is that they are mostly biased toward a specific vendor solution such as VMware vSphere or Microsoft Hyper-V and they do not provide a solid foundation in the key concepts surrounding virtualization.

As one of the largest professional training institutions with branches throughout the world, New Horizon Computer Training Center provides various virtualization training courses, but almost all the courses pertain to vendor specific solutions. For example, in a course called "VMware vSphere Install, Configure, Manage v5.1", various processes and tasks involved in installation, configuration, and management of VMware vSphere 5.1 are taught.

Another well-known professional computer training institution called Global Knowledge also provides various virtualization training courses. In a course called "Implementing and Managing Microsoft Server Virtualization", various aspects of Microsoft Hyper-V including installation, virtual machine creation and deployment, physical-to-virtual conversion, and virtual-to-virtual conversion are taught.

Although virtualization is used in many universities to set up laboratories for system-oriented courses, there are not many courses in universities on teaching the virtualization technology itself. In the article titled “The Role of Virtualization in Computing Education”, Gaspar states that system administration courses can be better comprehended through hands-on work. He further discusses the role of virtualization in the creation of laboratories to provide a practical environment that has not been possible before a virtualization solution became available.

To expand its educational offerings, RIT has established its campus branches in remote locations such as UAE, Dubai and other countries. One of the main issues in setting up networking, security, and system administration classes in remote branches was to create a hands-on environment through which practical work pertaining to the theoretical lectures could be performed. Taking into consideration the intention of the faculty of Networking and System Administration department to use the existing hardware resources at RIT New York, the idea for the creation of a virtual lab environment called “Remote Laboratory Emulation System (RLES)” came into existence. In an article called “The Development and Deployment of a Multi-User, Remote Access Virtualization System for Networking, Security, and System Administration Classes”, Border explains how, with the aid of virtualization and other standard industry technologies, RLES was created. Furthermore, although there is an equipped physical lab environment at RIT New York, it can be seen that RLES is used by both local and remote students as it provides acceptable levels of performance and ubiquitous access to a virtualization environment through standard web browsers.

The work of Gonzalez [26] utilizes Amazon EC2 Web Services to perform the hands-on labs related to a pre-requisite course called “Principles of System Administration” at the graduate level at RIT. In this work, Gonzalez basically follows a twofold objective. Students require an environment to practice system administration course work that is part of their normal course work and this environment is provided to them via Amazon EC2 Web Services. At the same time, students get insight about cloud computing services that enables them to understand and utilize the technology in their careers. This work is different than Gonzalez’s work in that it creates a completely new course with both lectures and hands-on labs. Apart from providing students with fundamental information regarding virtualization, the course teaches students about the top two virtualization solutions used by a majority of companies and organizations throughout the world. In this context, both Gonzalez’s work and this work are similar in that students are prepared to better fit their careers.

Chapter 1 - Introduction to Virtualization

Along with the advancements in the world of computing, use of the word “Virtual” has increased substantially. Online commercial applications have made creation of virtual shops possible. We can truly buy all the stuff that we can buy in physical shops through virtual shops. Virtual LAN (VLAN) is a logical grouping of network computers and resources that appear to be on the same LAN regardless of their geographical locations.

Computer virtualization is a virtualization technique that involves partitioning a physical server into a number of virtual machines using virtualization software. By virtualizing an object, we can basically obtain some greater measure of utility from the resource the object provides [6]. Although this research might refer to virtualization in various contexts, the focus of this research is on computer/server virtualization.

1.1. History of Virtualization

The official use of computer virtualization known as “Virtual Machine” dates back to August 2, 1972 with the existence of VM/370 to allow multiple computing environments to coexist on one physical system to save on the cost of expensive mainframes. VM/370 was the name of the operating system that enabled multiple users of IBM System/370 computing system to take advantage of separate and independent virtual machines that had the same architecture as the underlying IBM System/370. VM/370 was composed of three different operating systems, which were the Control Program (CP), the Conversational Monitor System (CMS), and the Remote Spooling and Communications

Subsystem (RSCS). These three operating systems together formed a general-purpose software suite and delivered the computing resources of the IBM System/370 machine to a wide variety of people and computers. CP was the operating system used on the physical computing machine (IBM System/370) to simulate multiple copies of the machine itself. “These copies, or virtual machines, were each controlled like the real machine by their own operating systems” [5]. In the context of current virtualization terms, CP was actually the hypervisor the same way VMware ESXi and Microsoft Hyper-V are. Although other operating systems could be used on each virtual machine, CMS was the typical operating system that supported the interactive use of a virtual machine by a person. Finally, RSCS was the operating system used to interchange information among machines linked with communications facilities. IBM System/370 model 158 is shown on the figure 2.

Figure 2. IBM System/370 Model 158

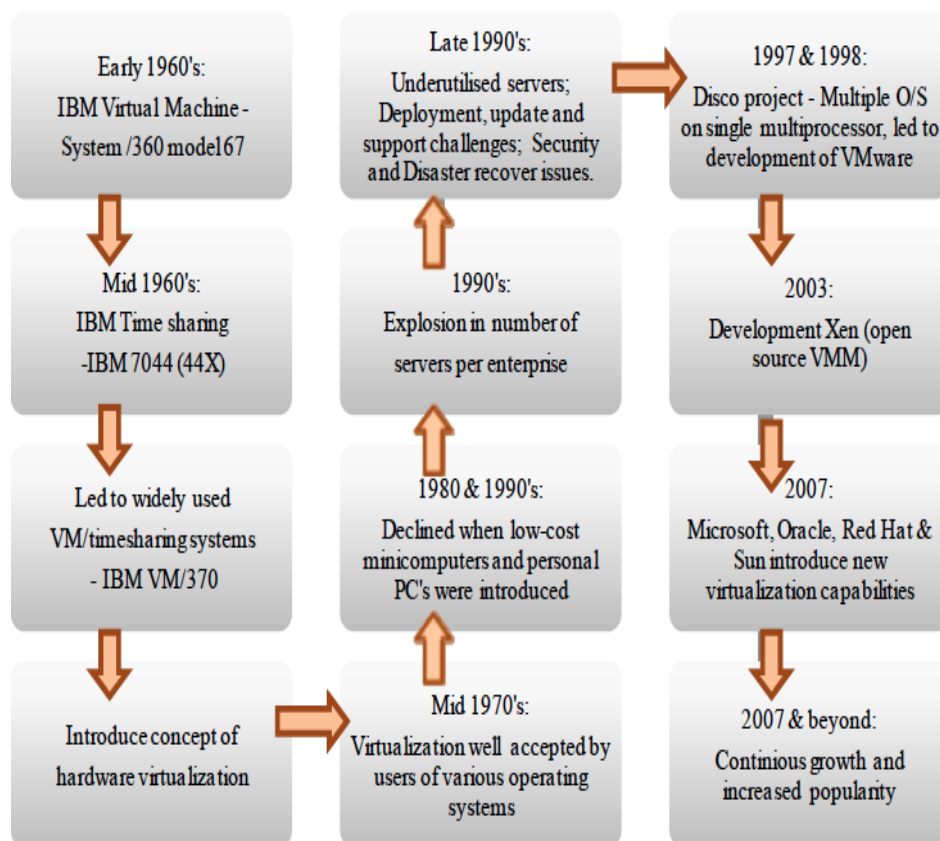


1.2. Distributed Computing Era

The use of virtual machines was popular in 1960s and 1970s to reduce the cost associated with mainframes and better utilize mainframes resources. Mainframes along with virtual machines enabled centralized computing. By the mid-1970s, virtualization was well established and organizations benefited from increased hardware performance (increased memory, system capacity, storage) through the use of virtualization. The use of virtualization declined with the introduction of low-cost x86 servers and desktops in 1980s and 1990s. During 1980s Microsoft developed Windows as a personal operating system and the user-friendly Microsoft Windows soon dominated other operating systems such as CPM and OS/2. With faster processors, an increased amount of memory, and other performance and capacity improvements in computer hardware resources, the Windows operating system was capable of running more powerful applications that had been run on minicomputers and mainframes. Being able to run powerful applications along with in-house expertise in companies allowed the Windows OS to dominate other operating systems. At the same time, since Microsoft Windows was designed to be a single user OS, it could run a single application, but installing a secondary application on the operating system could cause problems. The system requirements of each application could cause resource contention and failure of the operating system. On the other hand, different departments within a single company did not want to share any common infrastructure. Research and development, accounting, human resource, and other various divisions wanted isolated operating environments for themselves. For example, to secure the financial data of the company, the finance

department required a dedicated server. These policies could cause a company to use multiple servers to deploy a single application. The issues caused by installing more than one application on the operating system along with various corporate policies lead the application developers, designers, IT professionals, and vendors to adopt a “one server, one application” best practice [6]. Due to the reasons mentioned, the need for virtualization waved away in 1980s and 1990s and distributed computing was promoted. Figure 3 illustrates the evolution of computer virtualization.

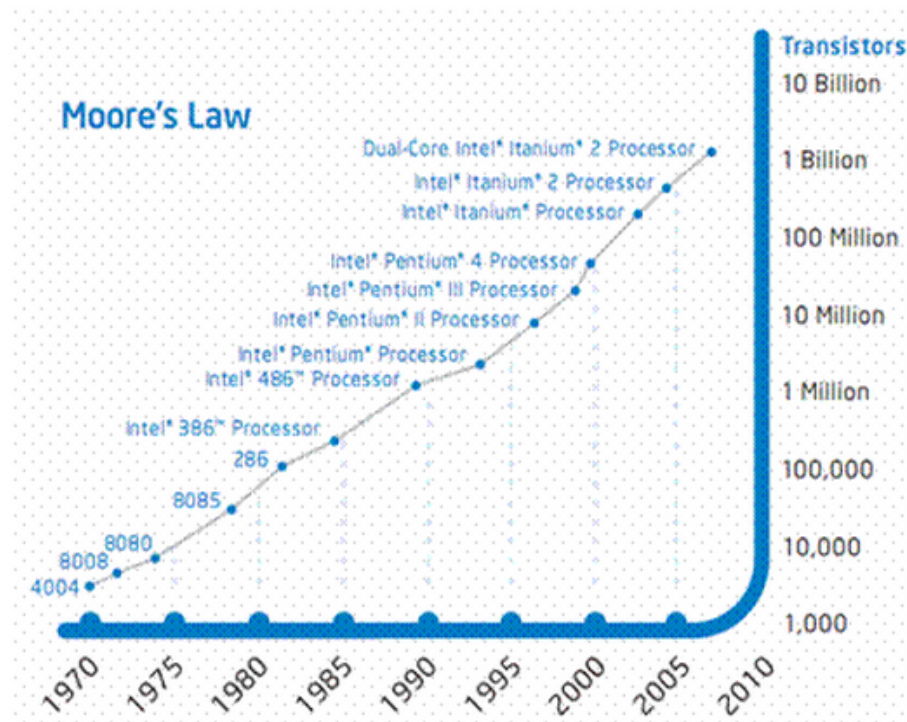
Figure 3. History and evolution of virtualization



1.3. Moore's Law

Named after Gordon Moore, one of the founders of Intel, Moore's law, publicized in 1965, states that the number of transistors on a chip will double approximately every two years [7]. Although Moore's Law was initially used for processors, it turned out to apply to other technologies such as memory and computer capacity. Servers are normally depreciated on a three to five year basis, which requires replacing the depreciated servers with new ones after three to five years of use. IT professionals traditionally purchase servers based on projected workload growth to avoid emergencies and procurement formalities in case more resource is required. This translates into putting a lot of computing power into a server, which may never be used by the application running on that server. Once the server is to be replaced, it is often replaced with a similarly configured model [6]. Taking into consideration that the server processing power doubles every two years according to Moore's Law, replacing a server after three years means a server with six times faster speed than the depreciated server. At the same time, if the server is depreciated after five years, it translates into a server with twelve times faster speed than the depreciated server. Finally, the application workloads running on the server do not normally increase at the same rate as server processing power. Overall the already oversized servers, which have been replaced on a three to five year basis have resulted in highly underutilized servers in physical environments (environments not using virtualization). Figure 4 depicts the transistor count from 1970 to 2010.

Figure 4. Moore's Law: transistor count



1.4. Challenges of Distributed Environments

A combination of “one server, one application” best practice and Moore’s Law has resulted in a number of challenges in physical computing environments including: server proliferation, low resource utilization, increased physical infrastructure costs, decreased scalability and agility, diminished disaster recovery, legacy systems, and migration. These challenges are discussed in the section that follows.

✚ **Server proliferation:** Also known as server sprawl, refers to underutilized server silos. Server proliferation is found to be the most common challenge in distributed environments and it can be as a result of

software vulnerability, “one server, one application” practice, scalability considerations, and other reasons [8]. Server proliferation can be considered the root cause for other challenges in distributed environments.

✚ **Low resource utilization:** Moore’s Law along with the “one server, one application” best practice have resulted in sever silos that are highly underutilized. As stated earlier, Windows servers on average utilize eight to twelve percent of physical server’s capacity and UNIX servers utilize 25 to 30 percent of physical server’s resources. Taking into consideration that Windows is the dominant operating system in data centers, one can envisage how underutilized the non-virtualized data centers can be.

✚ **Increased physical infrastructure costs:** As the number of servers increases, the costs associated with energy consumption, cooling, hardware component failures, support personnel, management overhead, software licenses, and other relevant costs increase. Overall, these costs increase the total cost of ownership (TCO) and decrease the return on investment (ROI) [8].

✚ **Decreased scalability and agility:** In the current dynamic and competitive world of business, all sorts of companies from small businesses to enterprises have to rapidly respond to new business initiatives and changes to remain in business. Depending on the companies policies and procedures on purchasing server equipment and service roll-out, the process of purchasing server equipment, testing the server hardware, installing the operating system and applications, and

patching and securing the server can take from a few weeks to a few months, which is a considerable challenge in today's business.

✚ **Diminished disaster recovery:** To a great extent, disaster recovery on physical servers is affected by the same scalability issues, but with a big difference in most cases. Delay in deploying a new service might not be noticed the same way failure of a previously deployed and used service is noticed. Purchasing new server equipment, testing the server hardware, installing the operating system and applications, and patching and securing the server can take some time. Once all these are done, the previously taken backups need to be restored, which further delays the process of bringing the service back online. On the other hand, keeping a redundant server for failover purposes increases the cost of hardware equipment, support personnel, data center space, and other relevant costs.

✚ **Legacy systems:** It is not uncommon for companies to depend on old unsupported applications, which only run on legacy operating systems, which in turn are only supported on legacy hardware. Keeping complete server failure out of thought, if not impossible, it is absolutely difficult to find the spare parts for obsolete server equipment.

✚ **Migration:** Migration from one server platform to another server platform can happen due to a number of reasons such as upgrading from an obsolete server platform to a brand new server platform. Migration in physical environments requires substantial amount of planning and considerations as the operating system and its application(s) are bound to the underlying hardware platform. At the same time, depending on the

workload and proper planning, for sure there will be some amount of downtime for the migration of associated workload.

The abovementioned challenges along with other issues in distributed environments led the IT industry toward the well-established solution used in 1960s and 1970s called virtualization.

1.5. Virtualization Terminology

In this section, a list of common virtualization terms along with their corresponding definitions are provided to create a clearer view of the terms used in the rest of this paper.

Virtual Machine

A virtual machine is a software implementation of a physical machine that can have its own operating system and applications.

Host (Host Machine)

A host machine is the physical machine running the hypervisor. The host machine provides the actual hardware resources such as CPU, memory, network connectivity, and other resources for virtual machines to run.

Hypervisor (Virtual Machine Monitor (VMM))

A hypervisor also known as Virtual Machine Monitor is a layer of software that makes virtualization possible. It abstracts the physical layer and presents this abstraction for virtual machines to use. It presents some subset of the physical resources to each individual virtual machine and handles the actual I/O from virtual machines to the physical device and vice versa. Hypervisors are the foundation for virtual environments and enable virtual machines to power up applications. Hypervisors are the core components in any virtualization solutions

and they are what allow multiple VMs to run on a single host to better utilize its hardware resources.

Guest Operating System

A guest operating system is an operating system that is installed and run in a virtual machine.

1.6. Virtualization Requirements

In a 1974 article called “Formal Requirements for Virtualizable Third Generation Architectures”, Popek and Goldberg describe the properties and requirements for a virtual machine and virtual machine monitor (VMM) that are still in use today and are considered as the prerequisites for any hypervisors.

Popek and Goldberg define three essential characteristics for a Virtual Machine Monitor (VMM).

- ✚ Equivalence: A VMM provides an environment for programs that is essentially identical with the original machine.
- ✚ Efficiency: Programs running in this environment show at worst only a minor decrease in performance compared to running the same programs on the original hardware based machine.
- ✚ Resource Control: A VMM is in complete control of system resources.

The first characteristic of the VMM is Equivalence, which is also translated as Fidelity. By Equivalence, they meant the program running under a VMM should have the same effect and behavior as that of the program running on the original physical host machine.

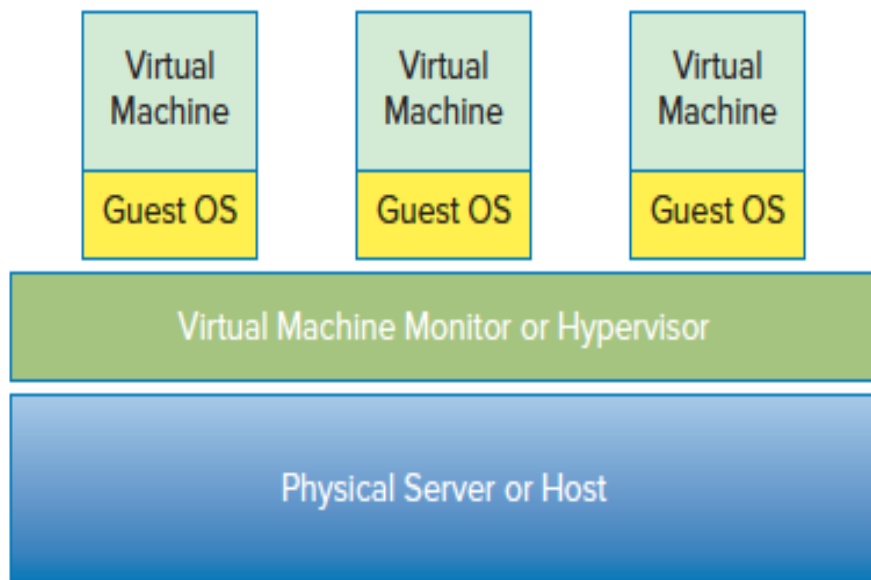
The second characteristic of the VMM is Efficiency, which is also translated as Performance. By Efficiency, they meant a statistically dominant subset of virtual

processor's instructions be run without any intervention by the hypervisor directly on the host physical processor. This requirement leaves aside the emulators and simulators (software interpreters) from the VM concept.

The third and final characteristic of the VMM is Resource Control, which is also translated as Isolation/Safety in the current virtualization terms. The VMM is said to be in complete control of the resources if two conditions are met:

1. A program running in an environment created for it does not have access to any resources not explicitly allocated to it.
2. The VMM can take control of the already allocated resources in certain circumstances.

Figure 5. A sample virtual machine monitor



Chapter 2 - Challenges in the Virtualization of the x86 Platform

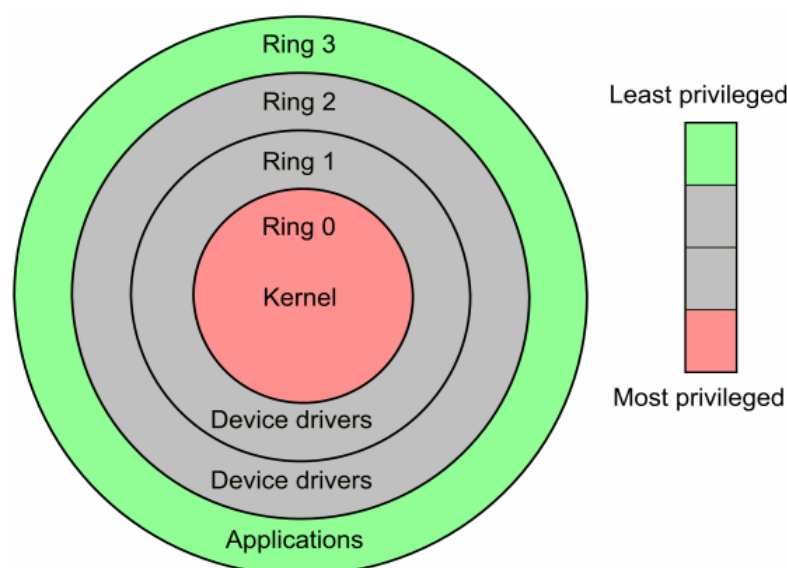
The Challenges associated with distributed environments in 1990s made researchers look to the possibility of building a virtual machine monitor on the dominant platform in data centers known as the “x86 platform”. There were two main challenges in the virtualization of the x86 platform. The First and biggest challenge was that the x86 platform would allow only a single operating system to execute privileged instructions in ring 0, which will be discussed in more details in the section that follows. The second challenge was to create a virtual machine monitor that would run guest operating systems in virtual machines without any modification to the guest operating systems themselves. This was a very important consideration as modifying the guest operating system offended the equivalence requirement mentioned in Popek and Goldberg article. At the same time, it also meant that the solution implemented in the virtual machine did not necessarily behave if it had run on the same physical counterpart [6].

The main issue with the virtualization of the x86 platform was that it was designed to run a single instance of an operating system, meaning it was not designed with virtualization support in mind. These processors provide a protection mechanism based on a 2-bit privilege level that defines what actions can be performed by which processes. The privilege levels, also known as rings, are numbered from 0 to 3 with ring 0 having the highest privilege level and ring 3 having the least privilege level. Figure 6 depicts these privilege levels.

The concept of privilege levels is used to provide a secure operating environment for x86 architecture through isolating user applications processes

from the operating system processes. Ring 0 with the most privilege provides unlimited access to the CPU and is where only the operating system kernel runs and controls access to the CPU. Ring 1 and 2 are rarely used, but traditionally these two rings are where the operating system's device drivers execute. Ring 3 with the least level of privilege is where the applications run. This architecture (privilege levels) ensures that an application running in ring 3 that is compromised cannot make privileged system calls [9]. In this privilege model, through limiting access to ring 0 to a single OS, the processor enables the OS to have complete knowledge of the state of the hardware. Taking into consideration that x86 processor architecture is designed to provide ring 0 access to a single operating system, x86 operating systems are designed accordingly and they consider themselves as the owner of the hardware. As a result, virtualization of the x86 architecture that provides the ability to run multiple operating systems on the same x86 platform requires placing an intermediary protocol called hypervisor between the virtual machines and the physical server platform to deliver shared resources of the physical server to various virtual machines.

Figure 6. Privilege levels for x86 platform



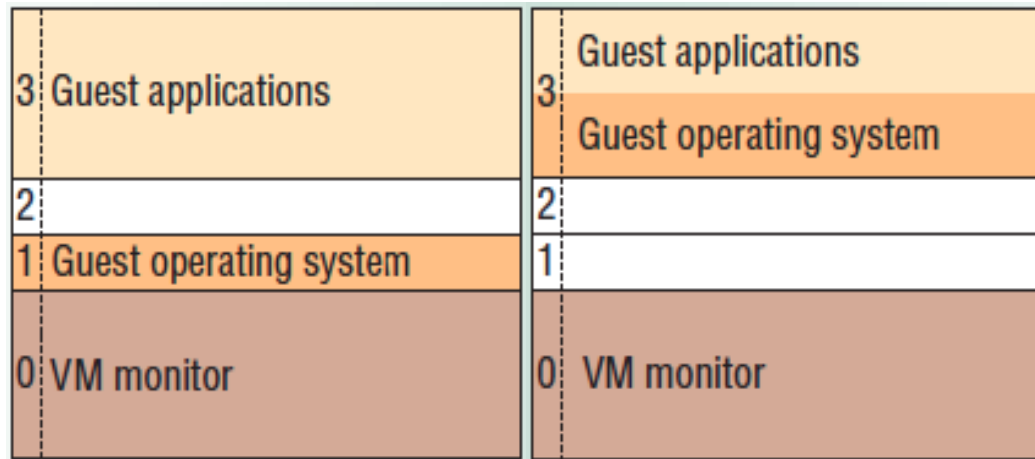
Once performing virtualization, the hypervisor must run at the most privileged level to control the hardware and system functions. Since the hypervisor runs in ring 0, no guest operating systems can function at this privilege level taking into consideration that x86 architecture limits ring 0 access to only one operating system. At the same time, there are some operating system's sensitive instructions (nonvirtualizable instructions) that can't be effectively virtualized as they have different semantics once they are not executed in ring 0. So the hypervisor needs to delude the guest operating system that it is also running in ring 0. But in reality, the hypervisor can't allow the guest operating system to run at this privilege level because doing so might corrupt the hypervisor code and data or provide the guest OS with access to privileged instructions.

2.1. Ring Deprivileging

Taking into consideration that the hypervisor can't allow the guest operating system to run in ring 0, the hypervisor moves up (deprivileges) the guest operating system to a ring above ring 0. There are two models to ring deprivileging, which are privilege level 1 (the 0/1/3 model) and privilege level 3 (the 0/3/3 model) [10]. If the virtualization solution uses the 0/1/3 model, the hypervisor deprivileges the guest operating system to ring 1. This enables the guest operating system to be in complete control of the applications running on top of it in ring 3 as ring 1 has higher privilege compared to ring 3. If the virtualization solution uses the 0/3/3 model, the hypervisor deprivileges the guest operating system to ring 3, where it runs in the same ring as its applications. Regardless of which ring deprivileging model is used by the

virtualization solution, the hypervisor always runs in ring 0. Figure 7 illustrates how the 0/1/3 model on the left and the 0/3/3 model on the right look like.

Figure 7. Ring deprivileging



Ring deprivileging is not challenge free and comes with its own sets of challenges. The hypervisor must constantly monitor the activities of the guest operating system to trap attempts to access the CPU and certain system calls (calls that are not virtualizable). Then it executes these calls by itself and emulates the results back to the guest operating system. Ring aliasing, address-space compression, silent privilege failure, and others are some of the challenges associated with ring deprivileging. For example, when software runs at a ring other the one for which it was written (operating systems are designed to run in ring 0, but through ring deprivileging they are moved to other rings), a problem known as ring aliasing can arise. Contrary to the guest operating system's belief that it is running in ring 0, in ring aliasing, the true privilege level of the guest OS is exposed. Executing a PUSH instruction on the CS register of the CPU that includes the current privilege level and then examining the result would reveal the privilege discrepancy. Another problem that can occur due to ring deprivileging is when the guest operating system thinks it has control of the CPU

state. It makes a valid request for the state of the CPU and the CPU state returned is the true state of the CPU controlled by the hypervisor, not the simulated CPU state of the guest operating system. These values are in conflict and can cause execution failure.

2.2. Virtualization Techniques

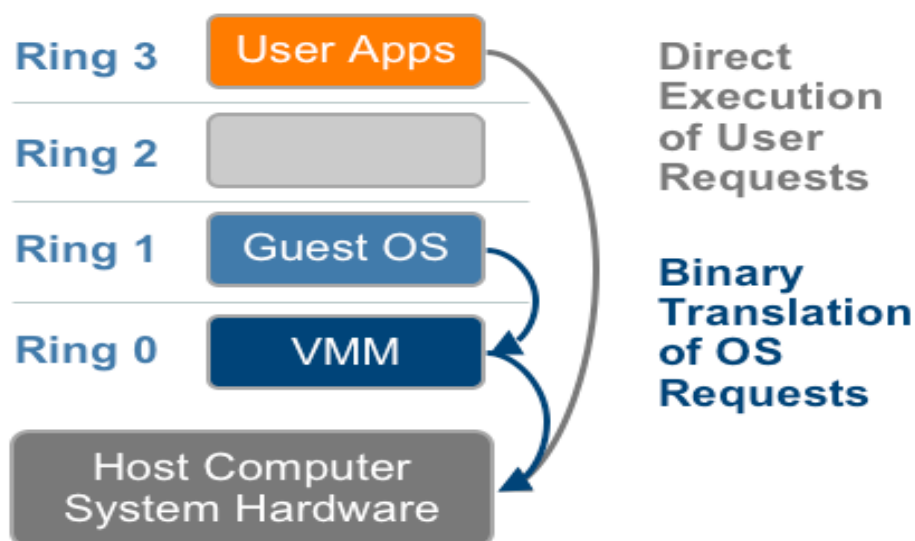
In order to correspond to the challenges of x86 virtualization (handling sensitive and privileged instructions to virtualize x86 architecture), there have emerged three main alternative virtualization techniques: “Full Virtualization using Binary Translation”, “OS assisted Virtualization or Paravirtualization”, and “Hardware Assisted Virtualization”. In the following section, each one of these solutions is explored.

2.2.1. Full Virtualization

As the leader of virtualization industry, once thought to be impossible, VMware figured out how to virtualize the x86 platform in 1998 using a technique known as Full Virtualization Using Binary Translation [11]. This technique, depicted in figure 8, utilizes the traditional direct execution with on-the-fly binary translation. In most modern operating systems, application programs run in privilege level 3 of the processor, which is virtualizable. Since user level applications are virtualizable, they run directly on the processor for high performance virtualization. At the same time, binary translation is used to translate the kernel level code to replace the nonvirtualizable instructions with new sequences of instructions that have the equivalent effect on the virtual

hardware [11]. The hypervisor scans the virtual machine's memory and traps the nonvirtualizable instructions before they are executed. The hypervisor then dynamically translates the instructions to the code with the same semantics once executed in a ring other than ring 0. The combination of direct execution along with binary translation results in full virtualization providing high performance virtual machine requiring no modification to the guest operating system or its applications and providing compatibility with a wide range of operating systems. A one to one mapping between the Popek & Goldberg virtualization requirements and the features of "Full Virtualization Using Binary Translation" technique, one can notice that this technique meets the requirements specified in Popek & Goldberg article. A majority of the virtual processor's instructions are directly executed on the physical processor. The virtual machine provides the guest operating system with a similar environment to the underlying hardware. Finally, similar to most hypervisors, the hypervisor is in complete control of the hardware resources.

Figure 8. Full Virtualization Using Binary Translation



2.2.1.1. Full Virtualization Characteristics

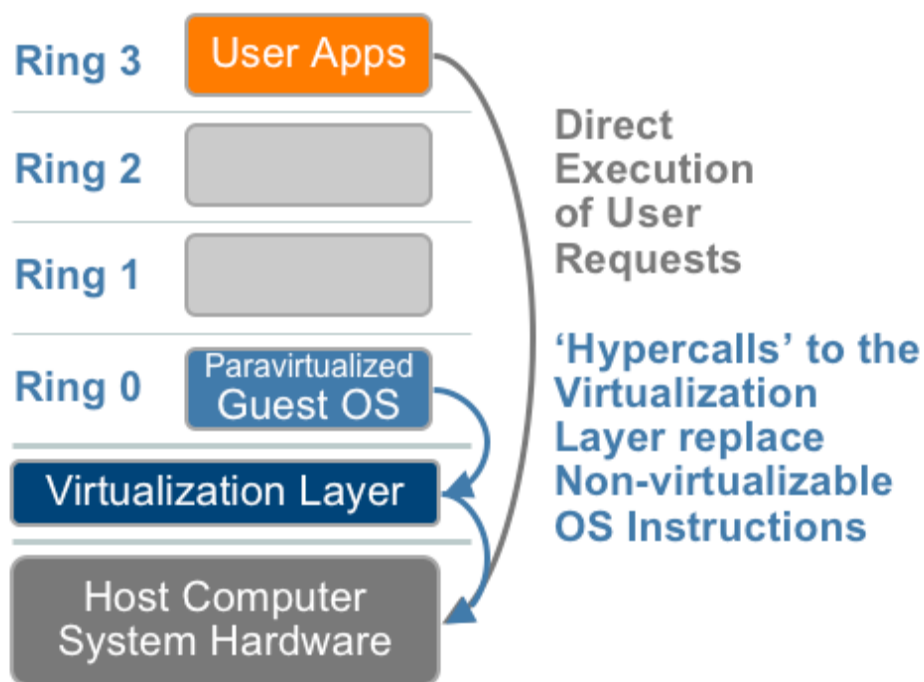
- ✚ Since there are no modifications required in any part of the guest operating system or its applications, the guest operating system is completely unaware that it is running in a virtualized environment.
- ✚ Since full virtualization uses direct execution on the processor with binary translation of privileged instructions, it neither requires any assistance on the part of the underlying hardware nor modification on the guest OS part.
- ✚ Since the guest OS has no idea it is being run in the virtualized environment, there is no communication between the guest OS and the hypervisor.
- ✚ Requiring no modification on the part of the guest operating system enables the hypervisor to support a wide range of operating systems.
- ✚ Binary translation of nonvirtualizable instructions requires highly sophisticated coding and consequently a great amount of effort on the side of the developer of the hypervisor.
- ✚ Hypervisors using binary translation come with higher performance overhead compared to hypervisors using paravirtualization because of the overhead related to dynamic translation of the nonvirtualizable instructions.

2.2.2. Paravirtualization

Well-known as OS assisted virtualization, illustrated in figure 9, paravirtualization is a virtualization technique that utilizes communication

between the hypervisor and the guest operating system to improve performance and efficiency. As the first paravirtualization solution, the Xen open source project was designed initially to support paravirtualized operating systems. In paravirtualization, the guest operating system's kernel code is modified so that the nonvirtualizable instructions are replaced with hypercalls that directly communicate with the hypervisor. The hypervisor also provides a hypercall interface for critical kernel operations such as memory management, interrupt handling, and time keeping [11]. So in order to solve the issue with nonvirtualizable instructions, these instructions are replaced with hypercalls that are sent to the associated interface on the hypervisor. A hypercall is similar to a Linux system call, which passes control into the hypervisor in ring 0. The hypervisor does the instruction and emulates the result back to the guest operating system. This means the hypervisor has to constantly monitor the guest operating system and trap the instructions that will fail.

Figure 9. Paravirtualization



2.2.2.1. Paravirtualization Characteristics

- ✚ Since the nonvirtualizable instructions are replaced with hypercalls through guest operating system modification, paravirtualization comes with lower virtualization overhead and better performance compared to binary translation.
- ✚ Although it is possible to modify open source operating systems such as Linux and OpenBSD, it is not possible to modify closed source operating systems such as Microsoft Windows. So paravirtualization does not support a wide range of operating systems.
- ✚ It is not practical to modify older versions of open source operating systems that are already in use.
- ✚ Paravirtualization requires changes to the guest operating system, which need to be implemented by the operating system vendor.
- ✚ Initially changes to the Linux operating system's kernel were made in the form of custom patches, but later starting with Linux kernel 2.6.23, the changes were incorporated into the mainline Linux kernel.

2.2.3. Hardware Assisted Virtualization

Considerable advantages of virtualization technology and high market demand for virtualization solutions induced chipset manufacturers (Intel and AMD) to produce virtualization aware processors that simplify virtualization. First generation hardware assisted virtualization for Intel and AMD were called Intel Virtualization Technology (VT-x) and AMD-V respectively. Both manufacturers provided a new CPU execution mode known as root mode that enabled the

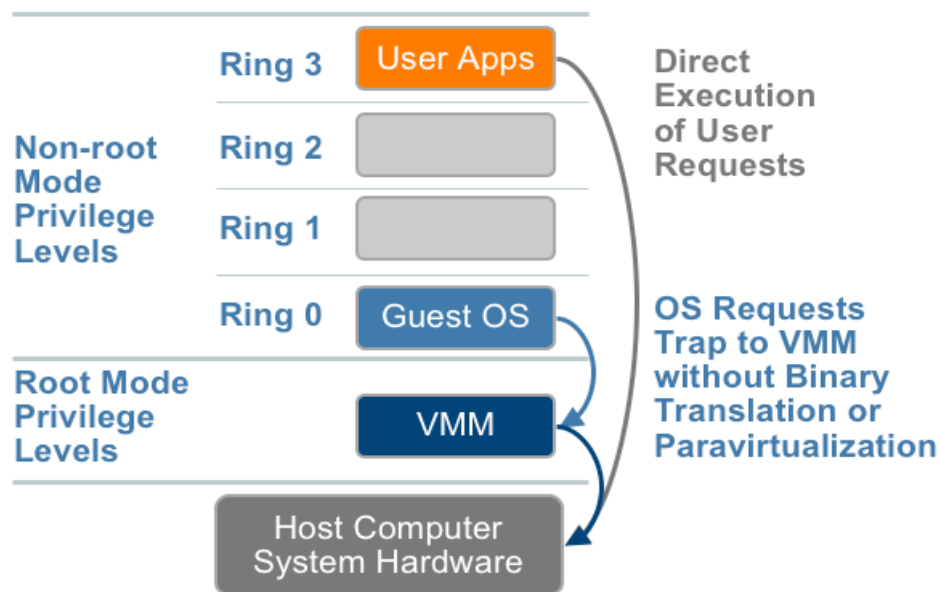
hypervisor to run in a layer below ring 0 to take control of the guest operating system. Using hardware-assisted virtualization, sensitive and privileged instructions automatically trap to the hypervisor removing the need for binary translation or paravirtualization [11]. Intel VT-x augments IA-32 with two new forms of CPU operations (two classes of ring), which are VMX¹ root operation and VMX non-root operation [10]. The hypervisor runs in VMX root operation to take full control of the CPU and other platform hardware. The guest operating system runs in VMX non-root operation because software running in the virtual machine must run at a reduced privilege level so that the hypervisor can take control of the platform resources. At the same time, both forms of the operation support all four rings and provide the hypervisor with the flexibility to use multiple privilege levels. The guest operating system runs within its expected ring levels and thinks it is in control of the CPU. Through hardware-assisted virtualization, the guest operating system is contained through VMX non-root operating level not through privilege levels. Figure 10 depicts hardware assisted virtualization.

Transitions between VMX root operation and VMX non-root operation are called VMX transitions. There are two kinds of VMX transition: Transition into VMX non-root operation is called VM entry (from hypervisor to guest operating system). Transition from VMX non-root operation to VMX root operation is called VM exit (from guest operating system to hypervisor). The Virtual Machine Control Structure (VMCS) is a new data structure that manages VM entries and VM exits and the processor behavior in non-root operations [10]. Furthermore, the guest operating system state is stored in VMCS. With the VM entry command,

¹ Virtual Machine Extensions

the guest operating system can execute VMX non-root operations. When the guest operating system passes control back to the hypervisor with the VM exit command, the hypervisor returns executing its privileged VMX root operations again [12].

Figure 10. Hardware Assisted Virtualization



2.2.3.1. Hardware Assisted Virtualization Characteristics

- ✚ Issue of the privileged instructions is sorted out by exiting to root mode.
- ✚ Provides excellent compatibility with large range of operating systems without requiring any modification in the guest operating system's kernel code.
- ✚ Second generation hardware assisted virtualization is required by most modern hypervisors such as VMware ESXi 5.5 and Microsoft Hyper-V 2012.

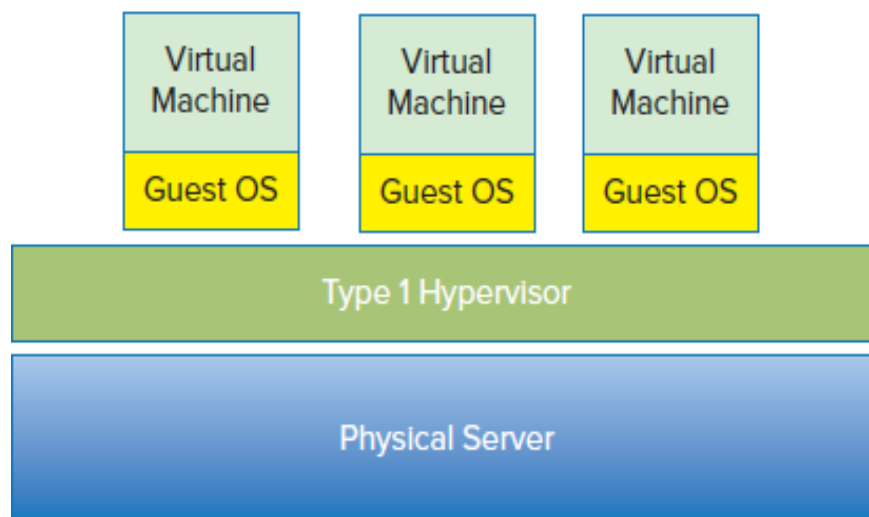
Chapter 3 - Types of Hypervisors

In the 1974 article called “Formal Requirements for Virtualizable Third Generation Architecture”, Popek and Goldberg classified two types of hypervisors called Type 1 (Bare metal) and Type 2 (Hosted).

3.1. Type 1 (Bare metal)

Type 1 hypervisor, illustrated in figure 11, is deployed as a bare-metal installation, which means that the hypervisor is the first thing installed on the host machine as the operating system. Type 1 hypervisor runs directly on top of the host machine to control access to the hardware and located right below the virtual machines to manage the guest operating systems.

Figure 11. Type 1 (Bare-metal) hypervisor



3.1.1. Type 1 Hypervisor's Characteristics

- ✚ The most important characteristic of type 1 hypervisor is performance. Because there is no intermediary layer between the hypervisor and the physical hardware, type 1 hypervisor directly communicates with the

physical hardware resources. This makes type 1 hypervisor a more efficient solution with better performance compared to type 2 hypervisor.

- ✚ Since type 1 hypervisors are written specifically to support virtualization, they usually have a very small footprint compared to general purpose operating systems. At the same time, designed specifically to support only virtualization, type 1 hypervisors enable us to provide most of the physical hardware resources of the host machine to the guest virtual machines. For example, VMware vSphere ESXi 5.5 has a footprint of less than 150 MB.

- ✚ The small footprint means less complicated software that usually results in more reliable and secure software.

- ✚ In general, type 1 hypervisors such as VMware ESXi are less compatible with hardware equipment due to their small footprint. To ensure compatibility with a virtualization solution, it is recommended that the Hardware Compatibility List (HCL) be checked to make sure the intended hardware components are compatible with the virtualization solution.

- ✚ One of the main purposes of computer virtualization is efficient use of hardware resources through consolidating multiple physical servers into one server in the form of virtual machines. The condensing of servers is called consolidation and is measured through a term called consolidation ratio. Consolidation ratio is calculated based on the number of VMs that can be fit on a physical server. For example, a server that has five VMs on it has a consolidation ratio of 5:1. Because type 1 hypervisors run directly on top of the host machine, they have considerably better performance and lower overhead compared to type 2 hypervisors. So the consolidation

ratios of this type of hypervisors are higher compared to type 2 hypervisors.

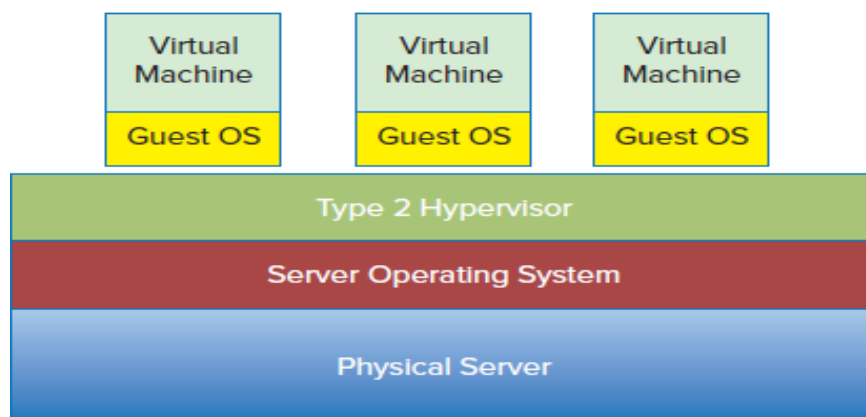
✚ Taking into consideration the performance and capabilities built into type 1 hypervisors, they are good candidates to run in data centers. These hypervisors are high performance with special capabilities (high availability, dynamic resource management) designed to support guest operating systems that require high availability features.

✚ VMware vSphere ESXi, Microsoft Hyper-V, Citrix XenServer, Red Hat Enterprise Virtualization (RHEV), and KVM are some instances of type 1 hypervisors.

3.2. Type 2 (Hosted)

Type 2 hypervisor, shown in figure 12, also known as hosted hypervisor is not deployed in bare-metal fashion. It is installed as an application on top of a traditional operating system. The type 2 hypervisor was the first x86 hypervisor offering because it could leverage the pre-existing operating system installed on the hardware for managing hardware resources. So it was the fastest way to introduce virtualization into the market.

Figure 12. Type 2 (Hosted) hypervisor



3.2.1. Type 2 Hypervisor's Characteristics

- ✚ Type 2 hypervisors are installed in the form of an application and most administrators are familiar with the process of installing an application on commodity operating systems such as Microsoft Windows and Linux.
- ✚ Type 2 hypervisors are more compatible with physical hardware devices compared to type 1 hypervisors. This is because this kind of hypervisors utilizes the hardware known and used by the underlying operating systems such as Windows and Linux.
- ✚ Because a type 2 hypervisor runs as an application on top of the operating system, there is an extra layer (the operating system) between the hypervisor and the hardware. This means every time that a VM performs read/write disk operation, network I/O operation, or any other hardware interactions, it has to give its request to the hypervisor similar to type 1 hypervisor. The difference with the type 1 hypervisor is that since the type 2 hypervisor does not have direct access to hardware, it has to go through one additional cycle and provide the request with the operating system, which handles the I/O. Once the request is processed, the operating system hands the result back to the hypervisor, which in turn passes the result back to the VM. So every transaction in a type 2 hypervisor requires two additional steps, which require more time and processing overhead compared to a type 1 hypervisor.
- ✚ Since type 2 hypervisors run atop the operating system, any issues such as malware, OS failure, bad device driver, and etc causing the underlying

OS failure, bring about the failure of all the virtual machines running on top of the OS.

✚ Taking into consideration the performance and limitations of type 2 hypervisors, they are not valid candidates to be used in data centers. At the same time, type 2 hypervisors can be used by application developers, who need to access a number of different operating systems on their local machines for test purposes.

✚ VMware Workstation, Microsoft Virtual PC, Parallels Workstation, and Oracle VirtualBox are some of the commercial and free type 2 hypervisors from leading virtualization vendors.

3.3. Hardware Virtualization Extensions

In recent years, virtualization has become such a compelling solution that induced the chipset manufacturers not only to respond to the issue of sensitive instructions, but also provide additional virtualization enhancements via their processors. In the following section, hardware virtualization extensions are discussed.

3.3.1. Memory Management Unit (MMU) Virtualization

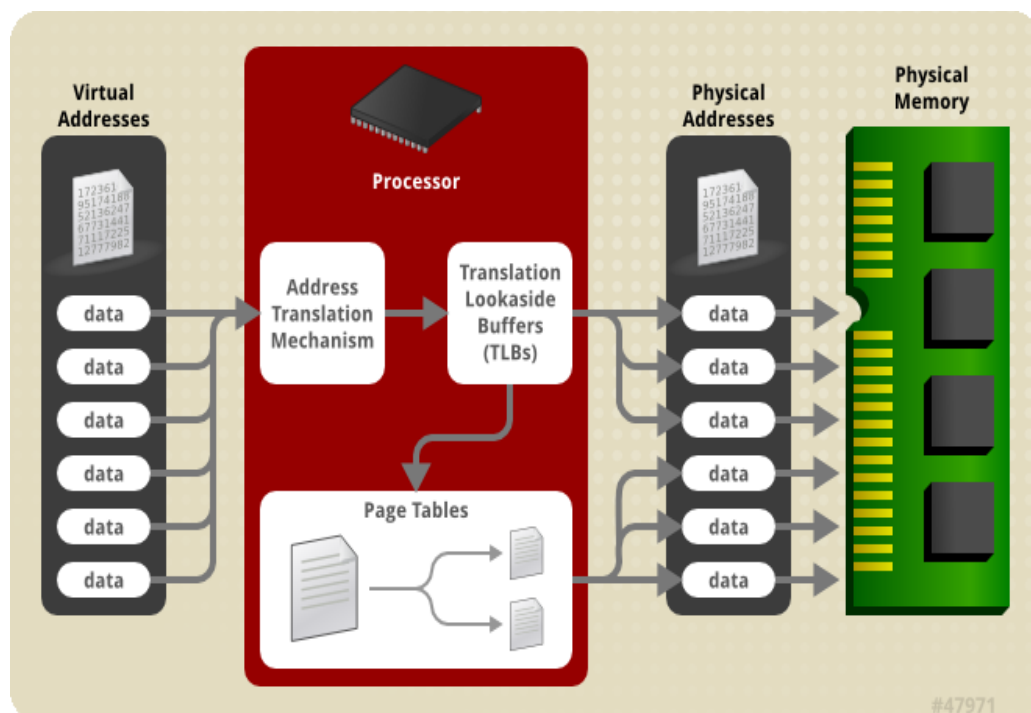
Memory Management Unit (MMU) Virtualization, also known as Second Level Address Translation (SLAT), is one of the CPU virtualization extensions that comes with both Intel and AMD processors. Although this section is about MMU virtualization, thorough understanding of MMU Virtualization requires

understanding address translation. So in this section, address translation is explained first followed by MMU virtualization.

The size and management of physical memory has been a challenging issue especially in cases shared among multiple tasks and different users. Although the size of physical memory has been increasing steadily during the past years, the amount of memory installed in most computers remains a limiting factor due to the enormous amount of memory required by various high memory demanding operating systems and applications running on computers [13]. To correspond to the issue of sharing limited memory by various applications and users, virtual memory has been created and it addresses the issue by hiding the presence of physical memory and instead presenting an abstract view of physical memory to applications. If more memory than what is available is required, the amount of accessible memory by various programs can be extended through using disk space. In this case, memory acts like a cache for the disk. The idea of extending memory by using disk space is called virtual memory. The issue with using the disk space for memory is that the disk is really slow to access. On the other hand, lots of disk space can be obtained at a small cost. Since the disk access is really slow, it should be avoided to access the disk unnecessarily. Operating systems use virtual memory mechanism to provide large address space to their applications. At the same time, virtual memory is used for memory protection by assigning every program a range of addresses called an address space. The main purpose of memory protection is to prevent a process from accessing parts of the memory that have not been allocated to it. This prevents a bug within a process from affecting other processes or the operating system itself. The operating system manages virtual address spaces and the assignment of real memory to

virtual memory. Utilizing special memory management hardware called Memory Management Unit (MMU), the CPU automatically translates virtual addresses to physical addresses. To improve virtual to physical translation speed, MMU uses a special cache called Translation Lookaside Buffer (TLB). TLB is implemented as a Content-Addressable Memory. Virtual addresses are used as keys to search the CAM (TLB) for equivalent physical addresses. If the requested address can be found in TLB, the search yields a match and the retrieved address can be used to access memory. This is called a TLB hit [14]. On the other hand, if the requested address cannot be found in the TLB, it is called a miss and translation proceeds by looking up the page table in a process called page walk. Since page walking is an intensive operation, all recent translations are stored in TLB to avoid subsequent page walks. Once the physical address is determined by page walk, the virtual to physical address mapping is placed into TLB for fast look up. Figure 13 shows the process of address/memory translation.

Figure 13. Address/Memory Translation



As mentioned, in the physical computing world, hardware techniques such as MMU and its cache are used to reduce the overhead associated with paging. However, in the virtualization world, the virtual machine's view of the physical memory is different than the host's view of the physical memory. As a result, a second level of translation is required to map the virtual machine's physical addresses to the host's physical addresses. To enable this additional translation level, two techniques based on software and hardware exist. The first technique that came into existence with the advent of the hypervisors is a software-based technique in which the hypervisor virtualizes the processor paging. As one of the main software-based virtualization techniques, shadow paging causes significant overhead in terms of reduced virtualization performance and increased CPU and memory utilization [15]. The other technique was born and implemented in hardware with the popularity of x86 hypervisor. Extended Page Tables and Rapid Virtualization Indexing are the memory management unit virtualization enhancements provided by Intel and AMD respectively. The first generation enhancements for processor virtualization came with Intel Virtualization Technology (VT-x) and AMD's AMD-V technologies, which only enabled CPU instruction set virtualization inside the processors. Second-generation processors from Intel and AMD included the support for Memory Management Unit Virtualization and came with Intel VT-i and VT-x and AMD-V technologies.

3.3.2. I/O MMU virtualization

I/O MMU is a memory management unit that manages computer I/O device access to the system memory. Similar to virtual to physical address translation done by MMU for computer software, I/O MMU performs mapping for DMA

capable I/O devices by mapping I/O device virtual addresses to physical addresses. I/O MMU can allocate large portions of memory to various I/O devices without the need for the portions of the memory to be contiguous. I/O MMU automatically maps the virtual addresses to the underlying corresponding fragmented physical addresses. At the same time, I/O MMU provides memory protection from I/O devices by enabling system software to control which areas of physical memory an I/O device may access. But this protection comes at the cost of additional direct memory access (DMA) overhead due to the required address resolutions and validations. To speed-up address resolutions, I/O MMU includes an input/output translation lookaside buffer (IOTLB).

I/O MMU virtualization in Intel CPUs is implemented by Intel VT-d technology and in AMD CPUs by AMD-Vi technology. In a virtualized environment, a hypervisor must be able to virtualize the I/O requests from the guest software. I/O virtualization can be supported through various models such as Emulation, Assignment, I/O Device Sharing, and New Software Interface [16]. In Assignment model, the hypervisor directly assigns the physical I/O device to the virtual machine. In this model, the driver for an assigned I/O device runs in the virtual machine to which the I/O device is assigned and the virtual machine directly interacts with the I/O device with minimal or no intervention from the hypervisor. Robust I/O assignment to virtual machines requires additional hardware support to ensure that the assigned device access is isolated and restricted to resources owned by the assigned partition. When an operating system is running in a virtual machine, it has no idea of what regions of host memory it accesses. This makes I/O device assignment, also known as Direct Pass-through, difficult because if the guest operating system attempts to perform

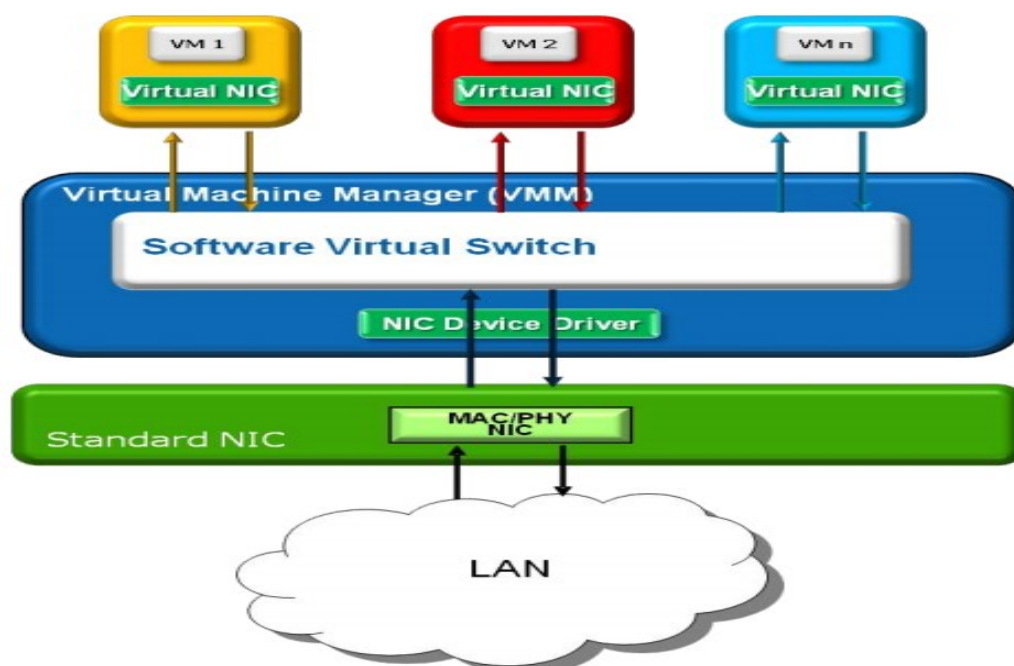
a direct memory access using its virtual-to-physical address mappings, it is possible that it corrupts the memory. In reality, the corruption is prevented by the hypervisor as it mediates the I/O operation to apply the translation that delays the I/O operations. I/O MMU virtualization provides the necessary translations and isolation in hardware, which improve performance and make pass-through possible.

3.3.3. Single Root I/O Virtualization

Developed by PCI-SIG, SR-IOV allows a PCIe device to appear as multiple separate physical PCIe devices [18]. I/O virtualization is basically a sharing of a single I/O device among multiple virtual machines. There are various models that I/O virtualization can be done, which are software-based sharing, hardware based sharing, and the hybrid model. Emulation in the software-based model, depicted in figure 14, is used to provide logical I/O hardware devices to the VMs. The emulation layer sits between the driver running in the guest OS and the underlying hardware to intercept all the relevant traffic issued by the guest driver. Emulation software then can parse the I/O commands, translate the guest addresses into host physical addresses, and then ensure that all referenced memory pages are present in the memory. Emulation software needs to resolve multiple I/O requests from all the virtual machines and send them serially into a single I/O stream that can be handled by the underlying hardware [19]. In this approach, significant CPU overhead may be required by the hypervisor to perform the required actions in software. Furthermore, some advanced features of the I/O device may not be available in this model as the software-based approach provides a subset of total functionality provided by the physical

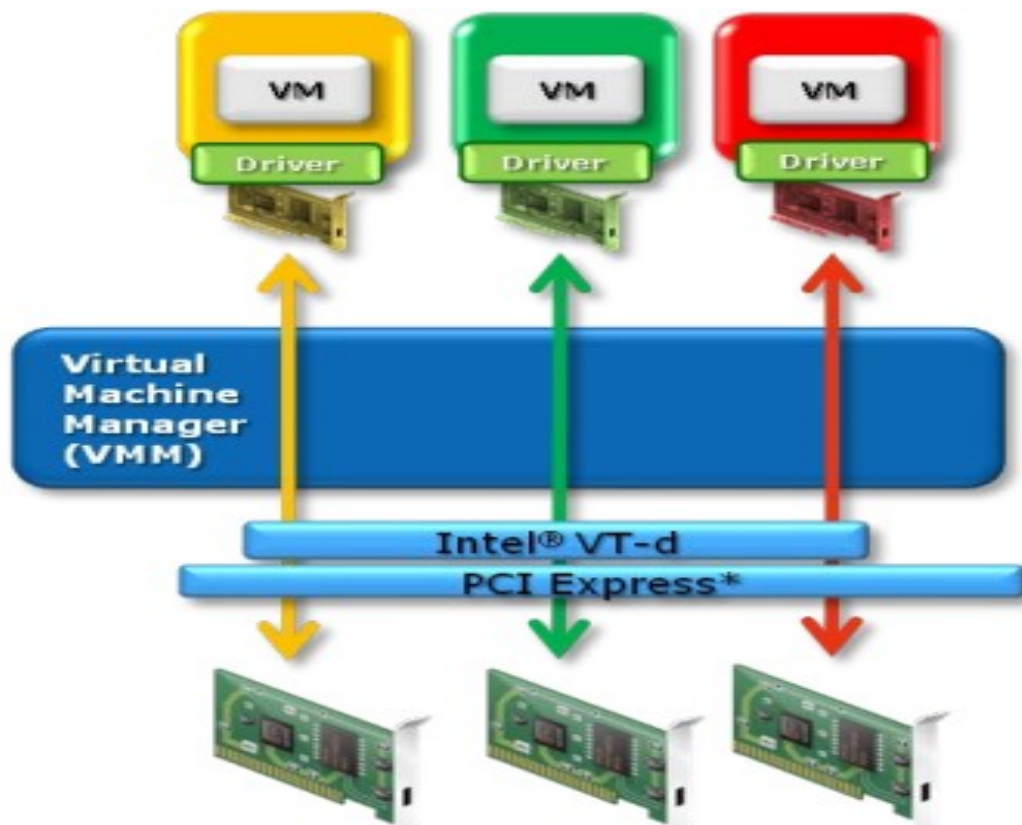
hardware. For example, the software-based approach eliminates the use of hardware acceleration that may be provided by the hardware device. Finally, the software-based approach creates extra overhead to each I/O operation due to the emulation layer that sits between the guest OS driver and the actual I/O hardware.

Figure 14. Software Based Sharing



Direct Assignment, illustrated in figure 15, is a hardware-based approach that works by directly exposing the hardware to the guest OS utilizing a native device driver. Discussed before, Intel VT-d is the technology that allows direct I/O device pass-throughs to the virtual machines through I/O MMU virtualization. However, this technique has limited scalability in that a physical device can only be accessed by one virtual machine. For example, a quad port NIC allows four direct assignments to only four virtual machines.

Figure 15. Direct Assignment (direct I/O device pass-through)



The goal of PCI-SIG SR-IOV specification is to standardize on a way of bypassing the hypervisor involvement in data movement by providing independent memory space, interrupts, and DMA streams for each virtual machine [19]. Taking into consideration the performance overhead of the software-based approach and scalability issues associated with Direct Assignment, there has emerged the need for the devices that are natively shareable. These devices replicate the necessary resources for each VM to be directly connected to the I/O device so that the main data movement occurs in hardware without any mediation from the hypervisor.

Since these natively shareable devices are accessed over PCI slot and can be implemented in standard and propriety ways, the PCI-SIG decided to create a standard approach called “PCI-SIG Single Root I/O virtualization and Sharing”.

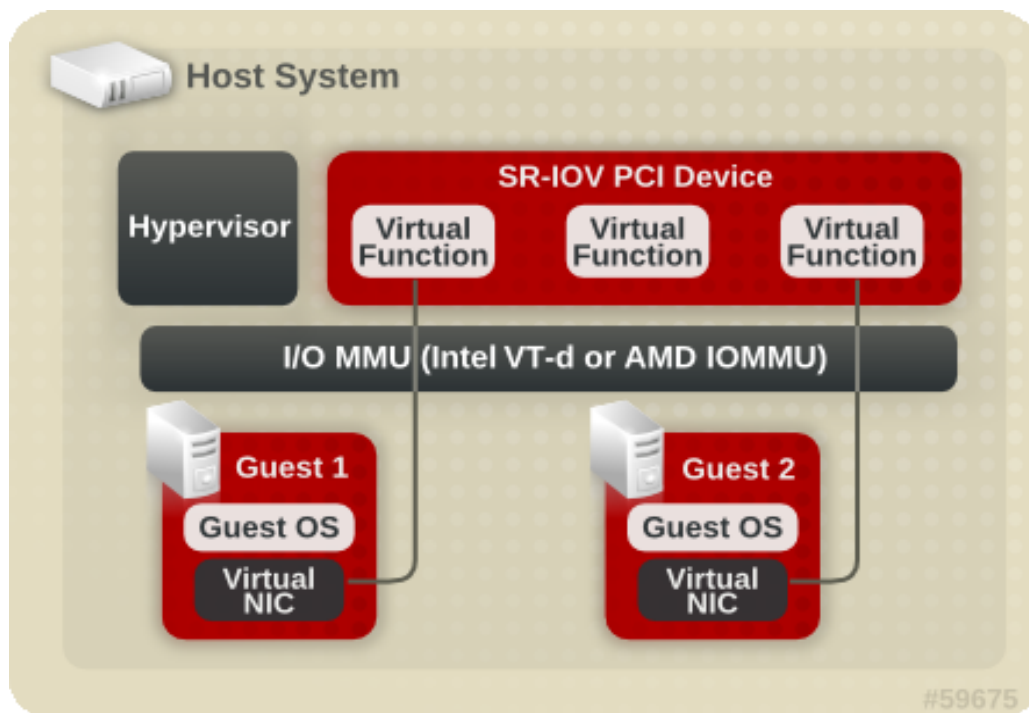
SR-IOV provides two new function types:

- 🚦 Physical Functions (PFs): PFs are fully featured PCIe functions that can be discovered, managed, and manipulated similar to any other PCI device. PFs have full configuration resources and can be used to configure and control the PCIe device.
- 🚦 Virtual Functions (VFs): VFs are lightweight functions that can't be configured. A Virtual Function shares one or more physical resources with the physical function and with other VFs that are associated with the same PF.

The PCI-SIG SR-IOV specification indicates that each SR-IOV device can have a PF and each PF can have up to 256 VFs associated with it. Although in theory each PF can have up to 256 VFs associated with it, because each VF requires the actual hardware resources from the physical device, the practical limits are much lower. At the time of this paper, 64 VFs seems to be the maximum for most devices. For example, a quad-port SR-IOV NIC presents itself as a single PCIe device with four ports. Each of these ports is a PF and each PF can have up to 256 VFs for a total of 1024 VFs for all 4 PFs (Physical ports). In reality, the number of VFs associated with each PF depends on the hardware resources of the physical device and how it is made. Finally, SR-IOV requires support both in the BIOS and operating system or hypervisor that is running the hardware. As mentioned before, PFs are full-feature PCIe functions that can be discovered and managed similar to any other PCIe devices. On the other hand, VFs are similar to PFs, but

lack the configuration resources. They have the ability to move data in and out. To avoid configuration changes on the underlying PF and resultantly all other VFs, configurations of VFs cannot be changed. Configuration can only be done on the PF. Since VFs are lightweight versions of PCIe, the OS or the hypervisor must be aware that they are not full PCIe devices. As a result, OS/hypervisor support is required for SR-IOV so that the OS/hypervisor can properly detect and initialize the PFs and VFs. Figure 16 shown below depicts Single Root I/O Virtualization.

Figure 16. Single Root I/O Virtualization



Chapter 4 - Virtualization Solution Leaders

In an article called “Magic Quadrant for x86 Server Virtualization Infrastructure”, Gartner evaluates primary virtualization solutions in terms of product/service, maturity, viability, sales, market responsiveness, operations, and other factors [4] and then provides ranking for each one of the solutions. The ranking, advantages, and disadvantages of each solution provide insight for companies planning to implement virtualization or might need to migrate to other virtualization solutions. Comparing the same article “Magic Quadrant for x86 Server Virtualization Infrastructure” via the graphs depicted in figure 17, 18, 19 for the years of 2011, 2012, and 2013, it can be easily noticed that VMware vSphere and Microsoft Hyper-V are the leaders of the virtualization industry and other solutions have not been able to keep pace with these two solutions. Although the other virtualization solutions surely have strong points and are being used in the industry, this paper only explores the architecture of the two leading virtualization solutions namely VMware vSphere ESXi and Microsoft Hyper-V in terms of I/O and memory management.

Figure 17. Magic Quadrant for x86 Server Virtualization Infrastructure (2011)

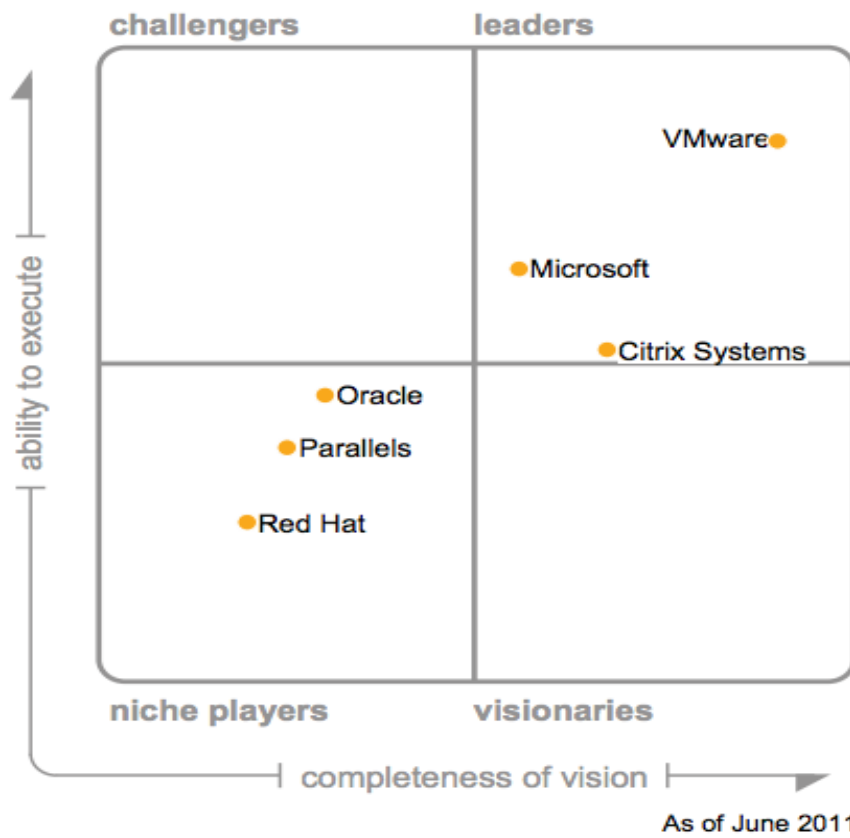
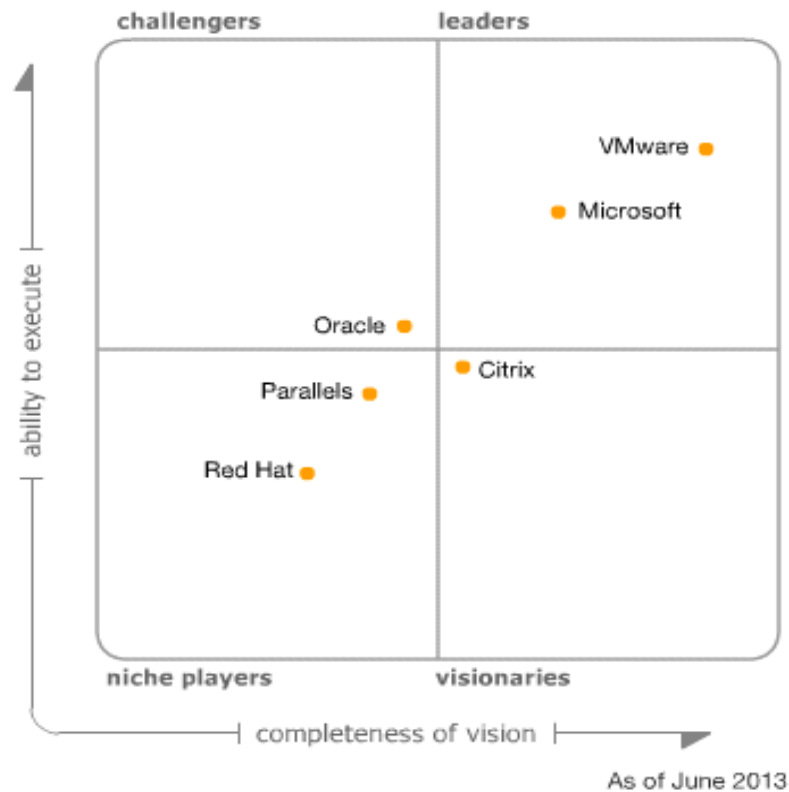


Figure 18. Magic Quadrant for x86 Server Virtualization Infrastructure (2012)



Figure 19. Magic Quadrant for x86 Server Virtualization Infrastructure (2013)



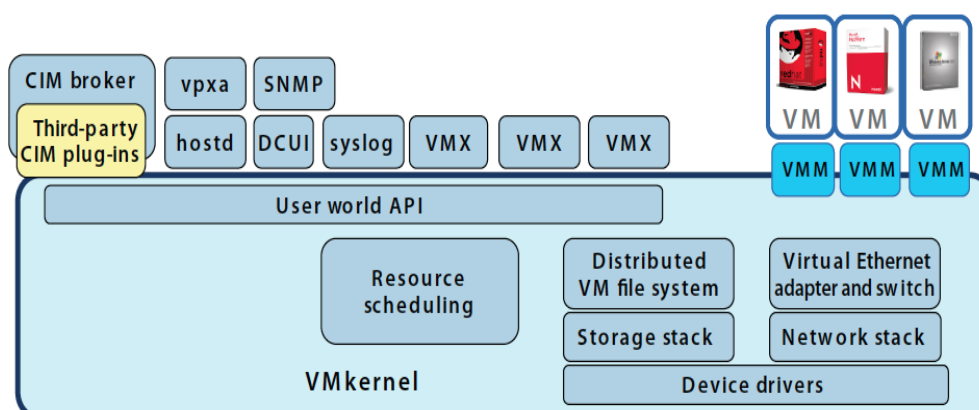
4.1. VMware vSphere ESXi

VMware ESXi is the slimmest hypervisor in the industry with less than 150 MB of install footprint. Contrary to its predecessor (ESX), ESXi runs without any aid from the Linux-based host operating system, which makes ESXi a more secure, easier to deploy, and easier to administer hypervisor. The functionality previously provided by the Linux-based host operating system, also known as service console, is replaced by remote command line interfaces in ESXi. At the very core of ESXi, there exists the VMKernel (the operating system) and a number of processes running atop VMkernel. VMkernel, shown in figure 20, is the foundation of the virtualization process in ESXi and manages the virtual machine's access to the underlying hardware resources by providing CPU

scheduling, memory management, I/O management, file system and other related functions [20].

There is a fundamental difference between VMware ESXi and most other hypervisors such as Microsoft Hyper-V and Citrix XenServer in that VMware ESXi handles I/O within the hypervisor itself. Since ESXi handles I/O within the hypervisor and does not depend on any general-purpose operating systems, it has greater throughput and lower overhead in terms of I/O management compared to other hypervisors. At the same time, since the I/O stack and device drivers are within the hypervisor itself [20], it has stricter hardware compatibility compared to Microsoft Hyper-V and other similar solutions. As a result, implementation of any VMware ESXi solution requires ensuring the hardware platform is compatible with ESXi before installation. On the other hand, both Citrix XenServer and Microsoft hyper-V depend on a parent partition known as dom0 for I/O management. For example, in the case of Hyper-V, the first component that is installed is the general-purpose Windows Server operating system (an example would be Windows Server 2012) and then Hyper-V role is installed. Microsoft Hyper-V can then utilize any hardware supported by the Windows Server operating system.

Figure 20. VMware ESXi Architecture



Following are the main processes running on the VMkernel [21]:

Direct Console User Interface (DCUI): Is a local, front-end, and keyboard-only interface that provides basic management and troubleshooting options should the ESXi host become inaccessible via the remote management tools such as vSphere Client or vCenter Server.

Virtual Machine Monitor (VMM): Is the process that implements the virtual machine hardware abstraction and is responsible for running a guest operating system [11]. Each virtual machine running on an ESXi host has its own VMM and VMX process.

Agents: There are various types of agents running on top of the ESXi host that are used for management of the ESXi host via remote applications.

Common Information Module (CIM): CIM is the interface that enables hardware-level management from remote applications via a set of standard APIs [21].

4.1.1. Memory Management in VMware ESXi

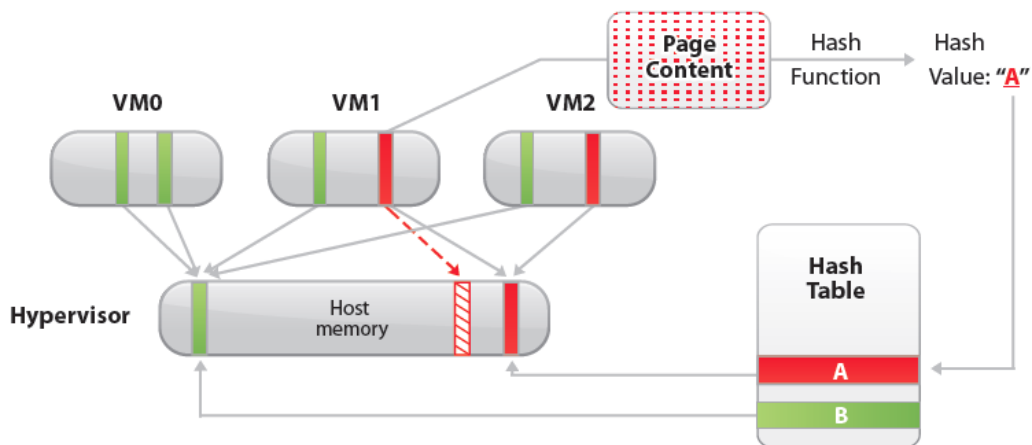
Similar to the physical environment that the operating system is responsible for allocating and reclaiming memory to and from applications, the hypervisor is in charge of allocating and reclaiming memory to and from virtual machines in the virtualized environment. ESXi can easily allocate a host's physical memory to a virtual machine as the first memory access of the virtual machine to host's physical memory generates a page fault that can be trapped by ESXi and resultantly allocate the memory. However, it is not easy for the hypervisor to know when to reclaim memory when the virtual machine is not using the host's physical memory. This is because the guest operating system's free list of

memory is not publicly available. ESXi supports memory over-commitment meaning the amount of guest physical memory of running virtual machines is larger than the amount of actual memory on the host machine [22]. Taking into consideration that memory is so critical in the operation of virtual machines, ESXi needs to perform memory over-commitment in such a delicate and efficient manner that performance of virtual machines is not affected or at most affected by a negligible amount. As a result, ESXi utilizes four memory management techniques to reclaim virtual machines' memory to provide efficient memory over-commitment. These techniques are transparent page sharing, ballooning, memory compression, and hypervisor swapping, which are further discussed below.

4.1.1.1. Transparent Page Sharing (TPS)

TPS, depicted in figure 21, works on the basis of sharing identical memory pages among multiple virtual machines to reduce the total number of memory pages required. Once there are multiple virtual machines running on an ESXi host, there is a good chance for them to have identical memory pages. This can be due to having the same guest operating system, applications, or user data. TPS creates hashes of the contents of host memory pages to identify identical memory pages. Once the identical memory pages are identified, ESXi reclaims all the identical pages except one that will be shared by virtual machines using the identical memory page. TPS is a default feature and runs whether or not the ESXi host is short on memory. Finally, TPS is completely transparent to virtual machines and does not penalize their performance.

Figure 21. Transparent Page Sharing (TPS)

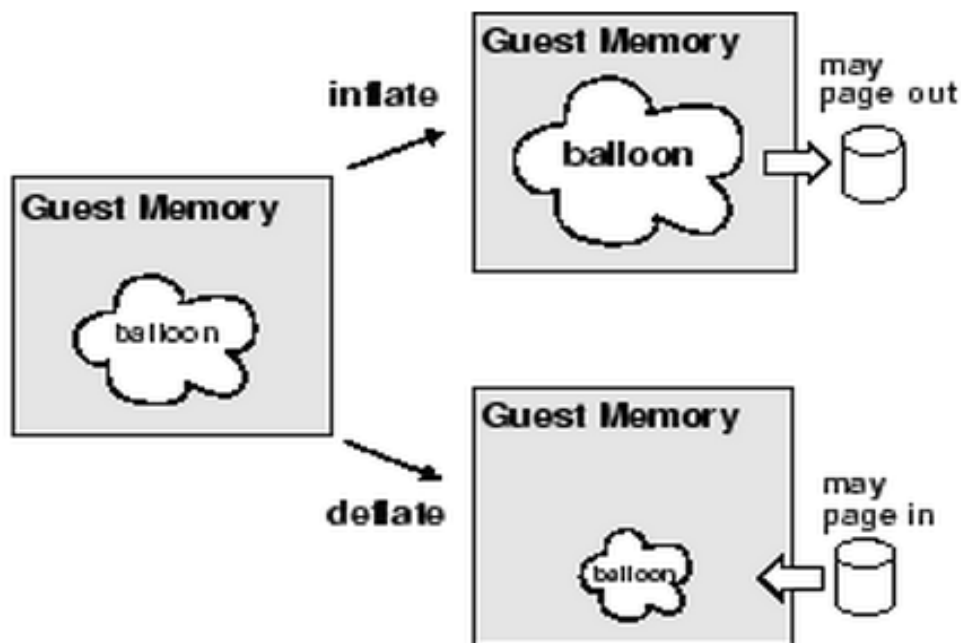


4.1.1.2. Ballooning

One of the key capabilities required for a hypervisor is isolation of virtual machines from one another. Isolation induces the guest operating system inside a virtual machine to be unaware that it is running in a virtualized environment and unaware of other virtual machines' states running on the same host. If there is memory pressure on the host due to excessive memory need by the virtual machines running on the host, the running virtual machines can't detect the memory pressure and will not free the guest physical memory. Ballooning makes the guest operating systems aware of the memory pressure on the host. It uses a driver installed as part of VMware Tools in the guest operating system. Once there is memory pressure on the host and the hypervisor requires reclaiming memory from virtual machines, the hypervisor sends a command to the balloon driver by specifying the balloon size and the driver responds to the command by reclaiming memory from the guest operating system. To reclaim memory from the guest operating system, the driver inflates and then passes over the collected memory pages back to the hypervisor for use by other virtual machines. Since

the balloon driver reclaims memory from the guest operating system (guest operating system decides which pages to free depending on the criticality of pages), if there is no memory pressure on the guest operating system, the hypervisor reclaims memory from the guest operating system without affecting its performance. On the other hand, if the guest operating system is already short on memory, balloon driver inflation can cause the guest operating system to page out guest physical memory to disk and affect the virtual machine's performance. Finally, once memory pressure on the hypervisor drops, the balloon driver deflates, which causes the memory to return back to the guest operating system. The main advantage of ballooning is that it allows the guest operating system to intelligently decide, which memory pages to be handed back to the hypervisor.

Figure 22. Ballooning



4.1.1.3. Memory Compression

Before moving to final performance-destructive memory management technique, VMkernel uses memory compression to compress memory pages and maintain them in a separate cache located in the main memory of the host. Compression is only performed on pages that can be compressed by at least 50 percent. Pages that can't be compressed by 50 percent are swapped out to disk. By compressing memory pages and keeping them in the fast-access host's memory, memory compression reduces the number of memory pages that are swapped out to disk and lowers the memory pressure on the host that is already under memory pressure. Memory compression comes into action only when the ESXi host is short on memory and ballooning has not been able to achieve the desired result. This technique is a default feature that can be disabled through the ESXi advanced settings.

4.1.1.4. Hypervisor Swapping

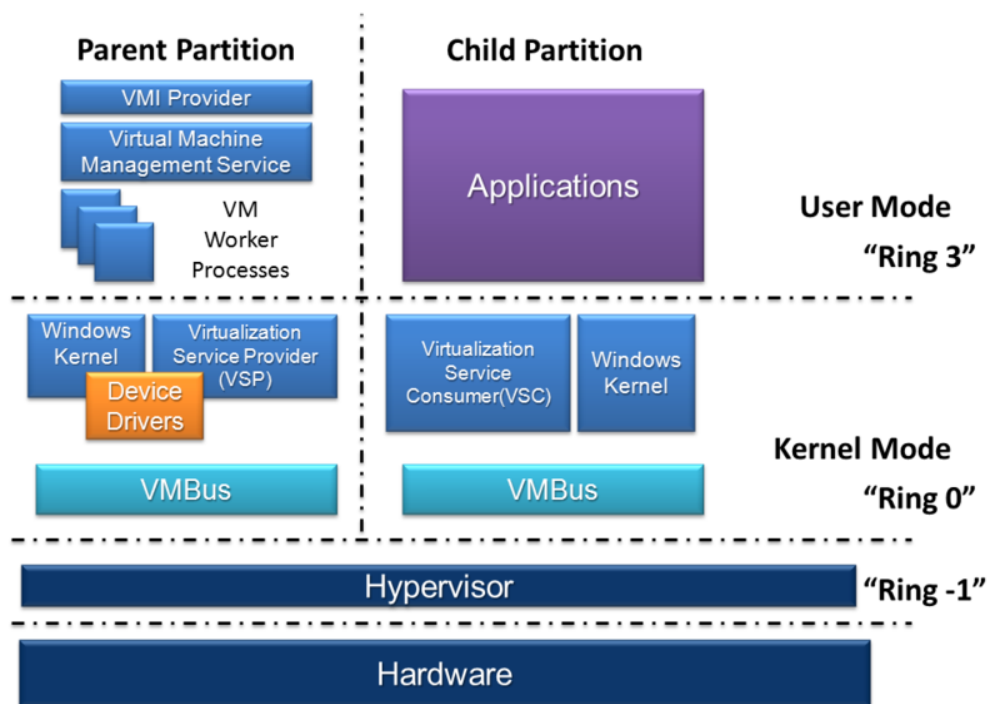
If the reclaimed memory pages by TPS, ballooning, and memory compression do not rectify the hypervisor memory pressure, ESXi will start swapping the virtual machine's memory to a virtual machine based swap file on disk to reclaim memory. In order to support swapping, the hypervisor creates a swap file for the virtual machine once the virtual machine is started. When memory pressure on the host is high and swapping is required, the hypervisor starts swapping the guest physical memory to the previously created swap file on the disk to free host physical memory for other virtual machines. Since hypervisor swapping happens without any knowledge of which memory pages are critical to the

operating system, the performance of guest operating system can be severely affected if critical pages of the guest physical memory are swapped. Finally, although TPS and ballooning take time to reclaim memory, hypervisor swapping is a guaranteed technique to reclaim a specific amount of memory within a specific amount of time [22].

4.2. Microsoft Hyper-V

Although Microsoft Hyper-V might be mistakenly considered as a type 2 hypervisor, it is a type 1 hypervisor that runs with the aid of a general-purpose host operating system (Windows Server 2012). This is because installing Hyper-V requires installing Windows server operating system first and then installing the Hyper-V role. Once Hyper-V role is installed, the host machine needs to be restarted so that Hyper-V is placed under previously installed Windows server operating system and the required changes are applied. Moving beneath the Windows server installation, Hyper-V runs in ring -1 and kernel of Windows server installation runs in ring 0 of the processor [23]. When Hyper-V is installed, the previous installation of Windows server operating system turns into what is known as Management OS or parent partition. Hyper-V depends on the Management OS for device drivers, I/O management, console access to host machine, and other operations. Figure 23 provides an overview of Hyper-V architecture.

Figure 23. Hyper-V Architecture



To provide isolated environments for virtual machines, Hyper-V uses the concept of partition. Each Hyper-V installation has a Management OS/parent partition and can have zero or more child partitions for virtual machines. The management OS is where the virtualization stack runs and has direct access to the host machine's hardware resources. The child partition does not have direct access to hardware resources and access requests to hardware resources are handled via the Management OS. The Virtualization Service Consumer (VSC) in the child partition sends requests to access hardware resources to the Virtualization Service Provider (VSP) in the Management OS via intercommunication channels known as the VMBus. Finally there are two Hyper-V related fundamental services running in the Management OS, which are Virtual Machine Management Service (VMMS.exe) and Virtual Machine Worker Process (VMWP.exe). VMMS is in charge of managing the state of virtual machines

running in child partitions. There is one VMWP for each virtual machine and the VMWP is responsible for interaction between virtual machines in the child partitions and the Management OS [24]. The VMWP is also responsible for tasks such as configuring, running, pausing, and snapshotting virtual machines.

4.2.1. Memory Management

Contrary to VMware ESXi, Microsoft Hyper-V does not support memory over-commitment. Microsoft uses the term Dynamic Memory for the memory management technology used in Hyper-V. Utilizing a driver installed in the guest operating system of the virtual machine, dynamic memory dynamically increases and decreases the amount of memory provided to the virtual machine based on the amount of memory required by the applications running on the virtual machine. As a result, dynamic memory enables efficient use of physical memory and a higher consolidation ratio for Hyper-V hosts. Dynamic memory in Hyper-V 2012 is only supported by specific guest operating systems and is composed of a number of fields that are briefly discussed below.

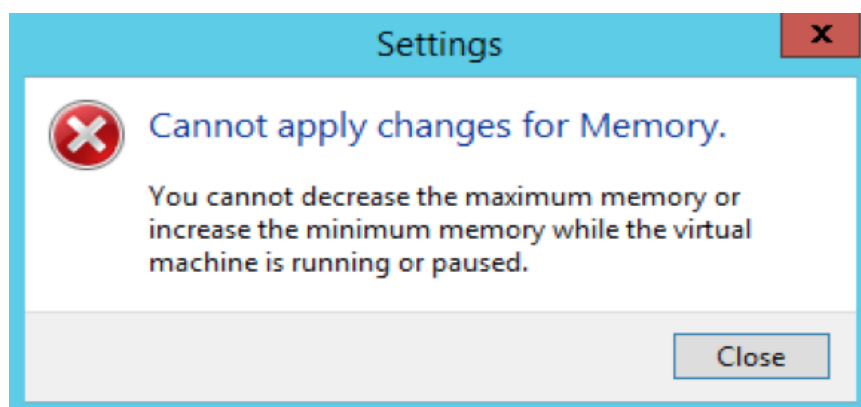
Startup RAM

Startup RAM specifies the amount of the host's memory that will be guaranteed to a virtual machine once it is started up and during the virtual machine's boot up. Hyper-V can't start a virtual machine if it can't allocate the amount of startup memory specified in the properties of the virtual machine. Startup memory should be at least set to the minimum amount of memory required by the guest operating system inside the virtual machine to boot up.

Minimum Memory

Minimum memory specifies the minimum amount of memory that must be allocated to the virtual machine after the guest operating system is booted up. In some scenarios such as Virtual Desktop Infrastructure, virtual machines require more memory to start up than once they are in steady state. Setting the minimum memory to a value lower than the startup memory enables dynamic memory to reclaim unused memory from idle virtual machines and better utilize physical memory of the Hyper-V host. Minimum memory can only be hot-decreased and the lowest available value for it is 32 MB. In order to increase the minimum memory, the virtual machine must be turned off. Increasing minimum memory on an online VM causes the error depicted in figure 24.

Figure 24. Increasing minimum memory error



Maximum Memory

Maximum memory specifies the maximum amount of memory that a virtual machine can use. Two conditions must be met for the virtual machine to use the amount of memory specified for the maximum memory. Firstly, the Hyper-V host must be able to provide the amount of memory specified in maximum memory field. Secondly, the guest operating system running in the virtual machine must

support the amount of maximum memory. The value can be set from as low as the value for Startup RAM to as high as 1 TB [25]. Maximum memory can only be hot-increased. In order to decrease the maximum memory, the virtual machine must be turned off.

Memory Buffer

Memory buffer is the extra amount of memory that is allocated to a virtual machine based on the amount of memory committed in the virtual machine. Since the amount of memory in the virtual machine using dynamic memory can vary, Hyper-V uses performance counters in the virtual machine to determine the actual amount of memory in the virtual machine to calculate the amount of memory buffer required for the virtual machine. Memory buffer is used in two scenarios. It is used as spare memory for the guest operating system if the instant need for memory by the guest operating system can't be fulfilled by dynamic memory. Windows operating system can also use memory buffer as file cache [23].

Memory Weight

Memory weight is used by dynamic memory to provide prioritized access to memory should there be contention on memory among the virtual machines running on the Hyper-V host.

Chapter 5 - “Virtualization Systems Architecture”

Similar to most other information technology courses, this eleven-lecture course is a combination of lectures and hands-on labs. Except the first lecture of the course, which provides students with fundamental knowledge about virtualization technology, all other lectures come with one or more hands-on labs to let the students better comprehend the theoretical lectures and prepare them for real-world scenarios. At the same time, the other ten lectures of the course are divided equally between the enterprise virtualization solutions of VMware and Microsoft companies. There are five lectures of VMware vSphere and five lectures of Microsoft Hyper-V. Although the whole course could be dedicated to VMware vSphere or Microsoft Hyper-V, as a university, RIT tries not to be biased toward any industry virtualization solutions. Furthermore, the course provides students with insight into the industry leading virtualization solutions so that students will make their own decisions, which virtualization solution to go for if they will need to do so. The following section briefly explores the content provided as part of each lecture.

5.1. Lecture1-Introduction to Virtualization

Educating students about the basic and fundamental knowledge is the basis of any university courses. Lecture one of the course teaches the students where virtualization technology comes from and how it has evolved. Similar to other technologies, virtualization came into existence in response to certain problems and there have been some challenges in development of practical virtualization solutions. This lecture also explains to students the difficulties of virtualization of

x86 architecture and the techniques used to cope with these difficulties. Finally, in this lecture, students get familiar with the types of virtualization solutions, examples of each type, and hardware extensions that have been developed to make virtualization solutions more efficient and practical. After finishing this lecture, students will learn about the followings:

- ✚ History of virtualization
- ✚ Requirements for virtualization software (Hypervisor)
- ✚ Reasons to use virtualization technology
- ✚ Challenges to virtualization of x86 architecture
- ✚ Types of hypervisor and their characteristics
- ✚ Hardware virtualization extensions

5.2. Lecture2-Introduction to VMware vSphere Infrastructure

This is an introductory course about VMware's enterprise virtualization solution known as VMware vSphere. VMware vSphere is not simple virtualization software, but a suite of comprehensive virtualization solution. Although students will not totally learn about all the products and features contained in the suite, they will get an overview of the products and features within the suite. Correct licensing is a very important factor in selection of virtualization software and keeping the companies legal. Student will learn how to correctly license their hosts with VMware vSphere. VMware vSphere deployment is beyond virtualizing physical machines. vSphere deployment affects various parts of the network infrastructure including storage, networking, and security, which should be given equal importance as in physical infrastructure. Planning and designing

considerations of VMware vSphere are briefly explained. After finishing this lecture, students will learn about the followings:

- ✚ Core components and features in VMware vSphere
- ✚ VMware vSphere licensing and packaging
- ✚ Planning deployment of VMware vSphere
- ✚ Installing VMware ESXi in interactive mode
- ✚ Connecting to a VMware ESXi host using remote management tools

5.3. Lecture3-Plan, Design, and Install vCenter Server

Enterprise class hypervisors require enterprise class applications that simplify deploying, managing, and maintaining virtualization environments. vCenter Server is an enterprise class application provided by VMware for the vSphere environment. Taking into consideration that proper planning, designing, and installing vCenter Server is absolutely critical to the operation of vSphere environment, students will be taught on planning, designing, and implementing vCenter Server based on VMware's best practices. After finishing this lecture, students will learn about the followings:

- ✚ What is vCenter Server and what its core services are
- ✚ Versions of vCenter Server
- ✚ Planning and designing vCenter Server
- ✚ Required components to install vCenter Server
- ✚ vCenter Server installation requirements and considerations
- ✚ Installing vCenter Server
- ✚ Using "vSphere Web client" to access vCenter Server

5.4. Lecture4-Designing, Creating, and Managing VMs in VMware vSphere

One of the primary reasons for the virtualization of physical servers is creating virtual machines and running applications on top of them. In this lecture, virtual machines are discussed from different perspectives. Students will learn about designing, creating, and managing virtual machines. At the same time, various ways that virtual machines can be created or deployed are further explained.

After finishing this lecture, students will learn about the followings:

- ✚ What a virtual machine looks like from inside and outside
- ✚ What files compose a virtual machine in VMware vSphere
- ✚ Designing, creating, and managing virtual machines
- ✚ Different ways to create/deploy virtual machines

5.5. Lecture5-Networking in the vSphere Environment

The same way networking is critical to the operations of a physical environment, networking is critical in a virtual environment. Networking in a vSphere environment is required to connect virtual machines to one another and to the physical network. Lecture five provides a solid knowledge in terms of differences and similarities between networking in a physical environment and a vSphere environment. It further teaches students about planning, designing, implementing, and securing networking in a vSphere environments. After finishing this lecture, students will learn about the followings:

- ✚ Networking components in vSphere environment
- ✚ Comparing physical and virtual switches
- ✚ Designing virtual networking in a vSphere environment

- ✚ Implementing VLANs in a vSphere environment
- ✚ NIC teaming in a vSphere environment
- ✚ Securing networking in a vSphere environment

5.6. Lecture6-Understanding High Availability in VMware vSphere

Placing several virtual machines on a single physical server is similar to placing all your eggs in one basket. A single point of failure is a serious issue with virtualization of physical machines. To compensate for this considerable issue, enterprise-class virtualization solution vendors have provided clustering and high availability solutions. This lecture provides an overview of high availability and the layers in which high availability can and should be considered. Then it talks about three different technologies that can be used to implement high availability in a vSphere environment, which are Microsoft Failover Clustering, vSphere High Availability, and vSphere Fault Tolerance. After finishing this lecture, students will learn about the followings:

- ✚ Layers in which high availability can be implemented
- ✚ Clustering in Microsoft Windows operating system
- ✚ Implementing Windows Server Failover Clustering in vSphere environment
- ✚ vSphere High Availability (HA)
- ✚ Distributed Resource Scheduler (DRS)
- ✚ vSphere Fault Tolerance

5.7. Lecture7-Introduction to Hyper-V

Similar to lecture two that provides an introduction to VMware vSphere, lecture seven provides an introduction to Microsoft Hyper-V. After finishing this lecture, students will learn about the followings:

- ✚ Microsoft Hyper-V architecture
- ✚ Types of virtual machine in Microsoft Hyper-V
- ✚ Licensing Microsoft Hyper-V
- ✚ Microsoft Hyper-V Features and Capabilities
- ✚ Microsoft Hyper-V best practices
- ✚ Microsoft Hyper-V Installation Requirements

5.8. Lecture8-Designing, Creating, and Managing VMs in Hyper-V

As mentioned earlier, creating and running virtual machines are among the first reasons to implement virtualization. In this lecture, students will learn to create, configure, and manage virtual machines in Microsoft Hyper-V. Various methods that can be used to migrate virtual machines from one host to another host are also explained. After finishing this lecture, students will learn about the followings:

- ✚ Creating virtual machines in Microsoft Hyper-V
- ✚ Configuring virtual machine settings
- ✚ Memory management in Microsoft Hyper-V
- ✚ Non-uniform Memory Access in Microsoft Hyper-V
- ✚ Types of virtual hard disks in Microsoft Hyper-V
- ✚ Virtual machine generations in Microsoft Hyper-V

- ✚ Live Migration

5.9. Lecture9-Networking in Hyper-V

Proper implementation of any virtualization solution requires an appropriate level of networking knowledge about that virtualization solution. Taking advantage of the networking lecture on VMware vSphere, this lecture explores the components involved in implementation of networking in Microsoft Hyper-V. It also teaches the students on how to plan, design, implement, and secure networking in Microsoft Hyper-V. After finishing this lecture, students will learn about the followings:

- ✚ Networking components in Microsoft Hyper-V
- ✚ Virtual switch extensibility
- ✚ Implementing VLANs in Microsoft Hyper-V
- ✚ NIC teaming in the Windows Server OS
- ✚ Single Root I/O Virtualization support in Microsoft Hyper-V
- ✚ Quality of Service

5.10. Lecture10-Understanding Hyper-V Replica

Virtualization technology has enabled new methods of disaster recovery and business continuity. Hyper-V Replica enables replication of virtual machines from a primary site to a disaster recovery site to enable prompt recovery in case any disaster occurs. This lecture pertains only to Hyper-V replica. After finishing this lecture, students will learn about the followings:

- ✚ What Hyper-V replica is and how it works
- ✚ Hyper-V replica considerations

- ✚ Methods to replicate virtual machines from a primary site to a disaster recovery site
- ✚ Hyper-V replica broker
- ✚ Monitoring virtual machine replication
- ✚ Failing over virtual machines

5.11. Lecture11-Building a Hyper-V Failover Cluster

The Comparison of VMware vSphere and Microsoft Hyper-V results in many similar features and capabilities due to similar customer requirements and the business competitiveness of VMware and Microsoft companies. One of the key features of any enterprise-class virtualization solution is high availability. Windows Server Failover Clustering is the high availability solution used for Microsoft Hyper-V that maps to the high availability feature in VMware vSphere. Due to the importance of high availability in a Hyper-V environment, this lecture is dedicated to the implementation of high availability using Windows Server Failover Clustering for Hyper-V. After finishing this lecture, students will learn about the followings:

- ✚ What Windows Server Failover Clustering is
- ✚ Failover clustering network prioritization
- ✚ Cluster Shared Volumes
- ✚ BitLocker
- ✚ Cluster-Aware Updating

5.12. Hands-on Labs

Similar to most information technology courses, all the lectures in this course except lecture one come in conjunction with hands-on labs so that students will better understand the theoretical lectures and acquire technical skills. The diagrams depicted in figure 25 and 26 pertain to the lab set-ups of the course.

All the hands-on labs provided as part of the course can be performed via these set-ups, which can be implemented in virtual or physical fashion.

Figure 25. Lab setup for VMware vSphere lectures

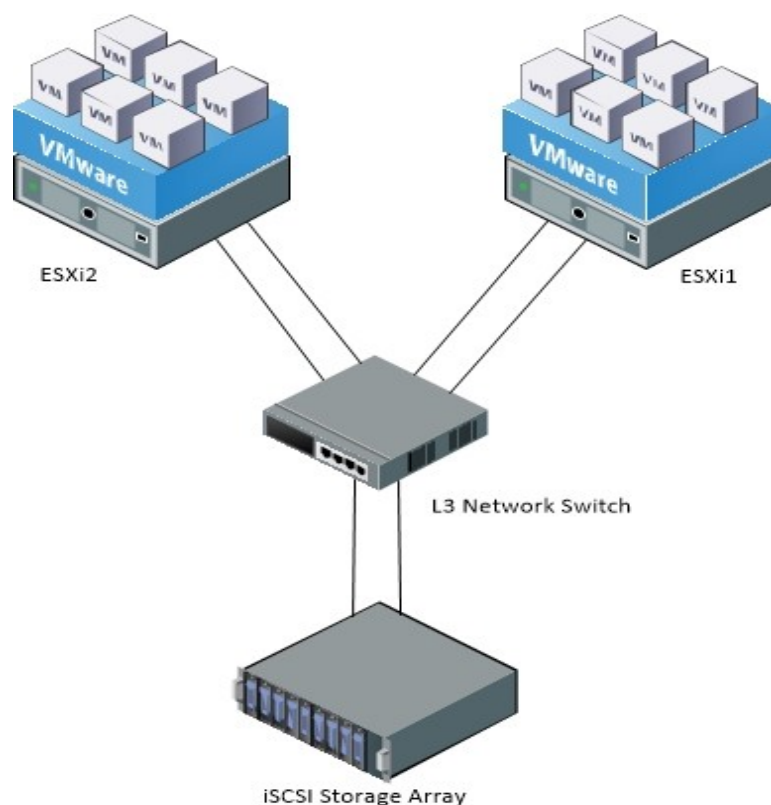
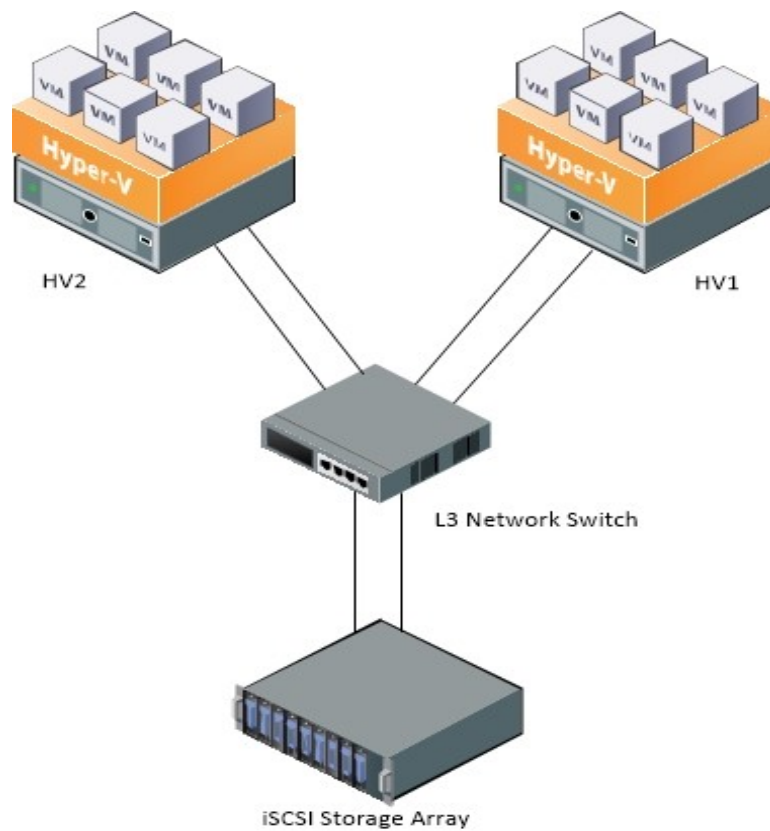


Figure 26. Lab setup for Microsoft Hyper-V lectures



Conclusion

The advantages of virtualization technology have turned virtualization into a standard technology in the network infrastructures of companies of many sizes. At the same time, by converting traditionally underutilized server silos into shared pools of resources, virtualization has established the foundation for a new processing model known as cloud computing. The rapid trend toward the utilization of virtualization and cloud computing by companies and organizations has brought about high demand for information technology professionals with virtualization and cloud computing expertise. The most commonly deployed and used virtualization solutions in the industry are VMware vSphere and Microsoft

Hyper-V respectively. Since the most commonly deployed virtualization solutions are provided by VMware and Microsoft, the main demand in the industry is for information technology professionals with expertise in VMware and Microsoft virtualization solutions. As one of the leaders in providing students with cutting edge technology, RIT puts up the course titled “Virtualization Systems Architecture” to better suit Networking and System Administration students in their careers. Apart from providing the students with fundamental information about virtualization technology, this course teaches students about planning, designing, and implementing a virtualized environment using the most commonly deployed virtualization solutions in industry namely VMware vSphere and Microsoft Hyper-V.

Appendixes

Appendix A – Virtualization Systems Architecture

The course “Virtualization Systems Architecture” will be delivered via power point presentation slides, hands-on labs, and other guides. For better readability, the power point presentation slides and hands-on labs are provided as separate files and only references to the titles of these files are provided hereunder.

Lectures of the Course

Lecture 1 - Introduction to Virtualization

Lecture 2 - Introduction to VMware vSphere Infrastructure 5.5

Lecture 3 - Plan, Design, and Install vCenter Server 5.5

Lecture 4 - Designing, Creating, and Managing VMs in VMware vSphere 5.5

Lecture 5 - Networking in vSphere Environment

Lecture 6 - Understanding High Availability in VMware vSphere 5.5

Lecture 7 - Introduction to Hyper-V 2012 R2

Lecture 8 - Designing, Creating, and Managing VMs in Hyper-V 2012 R2

Lecture 9 - Networking in Hyper-V 2012 R2

Lecture 10 - Understanding Hyper-V 2012 R2 Replica

Lecture 11 - Building a Hyper-V 2012 R2 Failover Cluster

Hands-on Labs of the Course

Lab2.1-ESXi-5.5-Interactive-Installation

Lab2.2-Create-A-VM-&-Install-GuestOS-In-A-VM

Lab3.1-Active-Directory

Lab3.2-vCenter-Server-Installation

Lab4.1-Management-of-Virtual-Machines-and-vApps

Lab4.2-Management-of-Virtual-Machines-and-vApps

Lab4.3-Creating-vApps-P2V-Conversion-ImportExport-OVF

Lab5-Creating-and-Configuring-vSwitches

Lab6-High-Availability-and-Business-Continuity

Lab7-Installing-Hyper-V-2012-and-Configuring-Basic-Settings

Lab8.1-Designing-Creating-and-Managing-VMs

Lab8.2-Designing-Creating-and-Managing-VMs

Lab9-Networking-in-Hyper-V-2012-R2

Lab10-Hyper-V-Replica

Lab11-Building-a-Hyper-V-Failover-Cluster

Appendix B – Course Proposal Form



ROCHESTER INSTITUTE OF TECHNOLOGY COURSE PROPOSAL FORM

ALISANO COLLEGE OF COMPUTING AND INFORMATION SCIENCES

Information and Science and Technologies

NEW COURSE: GCCIS- NSSA 714 Virtualization Systems Architecture

1.0 Course Designations and Approvals

Required course approvals:	Name/Chair:	Approval date:
Academic Unit Curriculum Committee		
Department Chair/Director Approval		
College Curriculum Committee		

Optional designations:	Approval date from appropriate committee:
<input type="checkbox"/> General Education	
<input type="checkbox"/> Writing Intensive	
<input type="checkbox"/> Honors	

2.0 Course information:

Course title:	<u>Virtualization Systems Architecture</u>
Short title: **	<u>Virtualization Systems Architecture</u>
Credit hours:	<u>3</u>
Prerequisite(s): ***	<u>NSSA 620 Emerging Computer and Network Technologies</u>
Co-requisite(s):	
Course proposed by:	<u>Pooriya Aghaalitari</u>
Effective date:	<u>Fall 2014</u>

	Contact hours	Maximum students/section
Classroom		
Lab		
Studio	3	24
Other (specify)		

2.a Course Information (check one)

X	New Course
	New Seminar Title
	Change to an Existing Course (please briefly explain the changes):

2.b Term(s) offered (check)

<input checked="" type="checkbox"/> X <input type="checkbox"/> Fall	<input type="checkbox"/> X <input type="checkbox"/> Spring	<input type="checkbox"/> Summer	<input type="checkbox"/> Other
---	--	---------------------------------	--------------------------------

All courses must be offered at least once every 2 years. If course will be offered on a bi-annual basis, please indicate here:

2. Student Requirements**Students required to take this course:**

None

Students who might elect to take the course:

MS students in NSA, SE, CS, IST, CSEC

3.0 Goals of the course:

Due to its considerable number of advantages, virtualization technology has become a standard in computing environments of different sizes. The purpose of this course is to educate students about the industry standard and commonly used virtualization solutions so that the students better suit their careers in the IT industry.

4.0 Course description:**GCCIS-NSSA-714 Virtualization**

This course establishes a solid foundation in terms of virtualization technology and the industry's leading virtualization solutions for students. A combination of lectures and hands-on labs, this course teaches students on how to plan, design, and implement a virtualized computing environment using industry standard and commonly used virtualization solutions namely VMware vSphere and Microsoft Hyper-V. Furthermore, students will learn about fundamentals of virtualization technology such as virtualization techniques, challenges of x86 platform virtualization, and why virtualization software is needed to perform virtualization.

(NSSA-620) Class 3, Credits 3, (Fall, Spring)

5.0 Possible resources (texts, references, computer packages, etc.)

Software: VMware ESXi, vCenter Server, Microsoft Hyper-V

References:

- Mastering VMware vSphere 5.5 by Nick Marshall, Scott D. Lowe
- VMware vSphere Design 2nd Edition by Forbes Guthrie, Scott Lowe and Kendrick Coleman
- Windows Server 2012 Hyper-V Installation and Configuration Guide by Aidan Finn, Patrick Lownds, Michel Luescher, Damian Flynn
- Virtualization Essentials by Matthew Portnoy
- VMware vSphere documents and VMware white papers
- Microsoft TechNet and online articles

6.0 Topics (outline):

1. Introduction to Virtualization
2. Introduction to VMware vSphere Infrastructure
3. Plan, Design, and Install vCenter Server
4. Designing, Creating, and Managing VMs in VMware vSphere
5. Networking in vSphere Environment
6. Understanding High Availability in VMware vSphere
7. Introduction to Hyper-V
8. Designing, Creating, and Managing VMs in Hyper-V
9. Networking in Hyper-V
10. Understanding Hyper-V Replica
11. Building a Hyper-V Failover Cluster

7.0 Intended course learning outcomes and associated assessment methods of those outcomes

Course Learning Outcome	Assessment Method
Describe the different models of virtualization utilized by major industry groups.	Exam
Plan, design, and implement a virtualized computing environment.	Lab Activity

8.0 Program outcomes and/or goals supported by this course (if appropriate)

- 8.1 Outcome 1: Describe technologies emerging in the field of networking and system administration and their impact on large organizations.
- 8.2 Outcome 2: Be a key contributing member in the development, management, or research of the computing infrastructure of an enterprise.
- 8.3 Outcome 3: Describe and implement technologies important to the management and deployment of large scale computing environments.

9.0

General Education Learning Outcome Supported by the Course (if appropriate)		Assessment Method
Communication		
	Express themselves effectively in common college-level written forms using standard American English	
	Revise and improve written and visual content	
	Express themselves effectively in presentations, either in spoken standard American English or sign language (American Sign Language or English-based Signing)	
	Comprehend information accessed through reading and discussion	
Intellectual Inquiry		
	Review, assess, and draw conclusions about hypotheses and theories	
	Analyze arguments, in relation to their premises, assumptions, contexts, and conclusions	

	Construct logical and reasonable arguments that include anticipation of counterarguments	
	Use relevant evidence gathered through accepted scholarly methods and properly acknowledge sources of information	
<i>Ethical, Social and Global Awareness</i>		
	Analyze similarities and differences in human experiences and consequent perspectives	
	Examine connections among the world's populations	
	Identify contemporary ethical questions and relevant stakeholder positions	
<i>Scientific, Mathematical and Technological Literacy</i>		
	Explain basic principles and concepts of one of the natural sciences	
	Apply methods of scientific inquiry and problem solving to contemporary issues	
	Comprehend and evaluate mathematical and statistical information	
	Perform college-level mathematical operations on quantitative data	
	Describe the potential and the limitations of technology	
	Use appropriate technology to achieve desired outcomes	
<i>Creativity, Innovation and Artistic Literacy</i>		
	Demonstrate creative/innovative approaches to course-based assignments or projects	
	Interpret and evaluate artistic expression considering the cultural context in which it was created	

10.0 Other relevant information (such as special classroom, studio, or lab needs, special scheduling, media requirements, etc.)

This course is intended to be run in one of the networking labs.

Appendix C – Course Syllabus

COURSE SYLLABUS

NSSA 714 VIRTUALIZATION SYSTEMS ARCHITECTURE

Class Time and Location:	TuTh 3:30-4:45 GOL 1610
Course Mode:	On-campus/Online
Prerequisite(s):	NSSA 620 Emerging Computer and Network Technologies

Instructor Information

Instructor:	Charles Border, Ph.D. Associate Professor Information Science and Technology
Contact Information:	Office: GOL-2329 Phone: 585-475-7946 Email: cbbics@rit.edu
Contact Policy and Preferences:	Office hours: MW 9:00 to 10:00, 11:00 to 3:00 Skype: charles.border
Online Course Material/Course Webpage:	The course materials will all be available through MyCourses

Course Description

NSSA-714 Architecture

Virtualization Systems

This course establishes a solid foundation in terms of virtualization technology and the industry's leading virtualization solutions for students. A combination of lectures and hands-on labs, this course teaches students on how to plan, design, and implement a virtualized computing environment using industry standard and commonly used virtualization solutions namely VMware vSphere and Microsoft Hyper-V. Furthermore, students will learn about fundamentals of virtualization technology such as virtualization techniques, challenges of x86 platform virtualization, and why virtualization software is needed to perform virtualization.

(NSSA-620) Class 3, Credits 3, (Fall, Spring)

Course Overview

This course will provide students with the core knowledge to plan, design, and implement a virtualized computing environment.

Course Learning Outcomes

Course Learning Outcome	Assessment Method
1. Describe different virtualization techniques and challenges of x86 platform virtualization	Assessed through homework and examinations
2. Describe VMware vSphere products, features, and licensing Install and configure VMware vSphere ESXi	Lab exercises and write-ups
3. Plan, design, and install vCenter Server	Lab exercises and write-ups
4. Design, create, and manage VMs in VMware vSphere	Lab exercises and write-ups
5. Plan, design, and implement networking in VMware vSphere	Lab exercises and write-ups
6. Implement and configure high availability for VMs in VMware vSphere	Lab exercises and write-ups
7. Describe Microsoft Hyper-V products, features, and licensing Install and configure Microsoft Hyper-V	Lab exercises and write-ups
8. Design, create, and manage VMs in Microsoft Hyper-V	Lab exercises and write-ups
9. Plan, design, and implement networking in Microsoft Hyper-V	Lab exercises and write-ups
10. Create, configure and manage Microsoft Hyper-V replica	Lab exercises and write-ups
11. Implement and configure high availability for VMs in Microsoft Hyper-V	Lab exercises and write-ups

Program Learning Outcomes

8.1 Outcome 1: Describe technologies emerging in the field of networking and system administration and their impact on large organizations.

8.2 Outcome 2: Be a key contributing member in the development, management, or research of the computing infrastructure of an enterprise.

8.3 Outcome 3: Describe and implement technologies important to the management and deployment of large scale computing environments.

Teaching Philosophy

I enjoy teaching classes based on problem based learning. I feel that students learn best when they can try technologies out to see how they translate into solutions for business problems. My goal in structuring my classes is to create an environment where student feel free to try new things and to try new and innovative things even at the risk of failure.

Audience

This course is meant for students who are interested in large scale commuting environments and the virtualization infrastructure that support them.

Course Topics

1. Introduction to Virtualization
2. Introduction to VMware vSphere Infrastructure
3. Plan, Design, and Install vCenter Server
4. Designing, Creating, and Managing VMs in VMware vSphere
5. Networking in vSphere Environment
6. Understanding High Availability in VMware vSphere
7. Introduction to Hyper-V
8. Designing, Creating, and Managing VMs in Hyper-V
9. Networking in Hyper-V
10. Understanding Hyper-V Replica
11. Building a Hyper-V Failover Cluster

Course Materials

Required Texts and Resources

- Mastering VMware vSphere 5.5 by Nick Marshall, Scott D. Lowe
- VMware vSphere Design 2nd Edition by Forbes Guthrie, Scott Lowe and Kendrick Coleman
- Windows Server 2012 Hyper-V Installation and Configuration Guide by Aidan Finn, Patrick Lownds, Michel Luescher, Damian Flynn
- Virtualization Essentials by Matthew Portnoy
- VMware vSphere documents and VMware white papers
- Microsoft TechNet and online articles

Resources will be linked to in MyCourses.

Media

As assigned

Required software:

VMware ESXi, vCenter Server, Microsoft Hyper-V

Course Schedule

Class meeting sections will be divided into lectures and labs. Monday and Wednesday classes will be devoted to lectures on new technologies while Fridays will be used for labs and student group work and will meet in the System Administration lab GOL 2320.

Option: Organize the activities in this table.

Week	Date	Topic/Activity	Readings/Discussions	Lab
1	1/27	Introduction to virtualization	Virtualization Essentials Chap 1, 2	No lab
2	2/3	Introduction to virtualization - continued	Virtualization Essentials Chap 1, 2	No lab
3	2/10	Introduction to VMware vSphere Infrastructure	Mastering VMware vSphere 5.5 Chap 1, 2	Lab 2.1 & 2.2
4	2/17	Plan, design and install vCenter Server	Mastering VMware vSphere 5.5 Chap 3	Lab 3.1 & 3.2
5	2/24	Designing, creating and managing VMs in VMware vSphere	Mastering VMware vSphere 5.5 Chap 9, 10	Lab 4.1, 4.2 & 4.3
6	3/3	Networking in vSphere environment	Mastering VMware vSphere 5.5 Chap 5	Lab 5
7	3/10	Understanding High Availability in VMware vSphere	Mastering VMware vSphere 5.5 Chap 7	Lab 6
8	3/17	Introduction to Hyper-V	Windows Server 2012 Hyper-V Installation and Configuration Guide Chap 1, 2	Lab 7
	3/24	Spring Break!!!!		No lab
9	3/31	Designing, creating, and managing VMs in Hyper-V	Windows Server 2012 Hyper-V Installation and Configuration Guide Chap 3	Lab 8.1 & 8.2
10	4/7	Networking in Hyper-V	Windows Server 2012 Hyper-V Installation and Configuration Guide Chap 4	Lab 9
11	4/14	Understanding Hyper-V Replica	Windows Server 2012 Hyper-V Installation and Configuration Guide Chap 12	Lab 10
12	4/21	Building a Hyper-V Failover Cluster	Windows Server 2012 Hyper-V Installation and Configuration Guide Chap 8	Lab 11
13	4/28	Open discussion		No lab
14	5/5			Practical exam
15	5/12	Review for final		No lab

16			Exams	
		5/23 Commencement		

Note any breaks, holidays or planned absences (such as for conferences) during the semester.

Grading/Evaluation

Your overall evaluation is based on the following components:

Class Participation	10%
Mid-term exam	20%
Assignments (3)	10%
Lab Write-ups/practical	25%
Lab attendance	10%
Final exam	25%
Total	100%

Grade Scale

Based on the 100% total listed above, letter grades will be assigned as follows:

A: 90 points or above B: 80 points to 89.9 points C: 70 points to 79.9 points

D: 65 points to 69.9 points F: below 65 points I: incomplete

Late Work

Assignments are due when assigned. Please let me know if an assignment is going to be late.

Attendance and Participation

There is a positive correlation between attending class and doing well in the class. Don't fall behind and don't blow off class.

Expectations

From students

I expect you to come to class prepared to learn and interested in the subject of our course. This will be an interesting class, but it will only be fun if you make it that way. Do the outside reading, take notes in class, don't expect to learn everything on one review of the material.

Time commitment

Since this is a three-credit hour course, you should plan to spend two hours per week in class, two hours in lab, and an additional six to twelve hours on readings, research, discussions, lab write-ups, assignments, etc. The rule-of-thumb is two to three hours per week outside the "classroom" for every credit hour per week in the classroom. If you do the math, it adds to twelve–sixteen hours per week, total.

Writing standards

Written work should adhere to Standard American English. Please proof your papers and e-mail messages before submitting them. I will grade for content, completeness, organization, spelling, grammar, and punctuation, as well as demonstration of knowledge gained in the course and your ability to apply it.

Course Policies

Academic Integrity Statement

As an institution of higher learning, RIT expects students to behave honestly and ethically at all times, especially when submitting work for evaluation in conjunction with any course or degree requirement. The Department of Information Science and Technology encourages all students to become familiar with the [RIT Honor Code](#) and with [RIT's Academic Honesty Policy](#).

Statement on Reasonable Accommodations

RIT is committed to providing reasonable accommodations to students with disabilities. If you would like to request accommodations such as special seating or testing modifications due to a disability, please contact the Disability Services Office. It is located in the Student Alumni Union, Room 1150; the Web site is www.rit.edu/dso. After you receive accommodation approval, it is imperative that you see me during office hours so that we can work out whatever arrangement is necessary.

Other Elements

Changes to the syllabus

I have provided this syllabus as guide to our course and have made every attempt to provide an accurate overview of the course. However, as instructor, I reserve the right to modify this document during the semester, if necessary, to ensure that we achieve course learning objectives. You will receive advance notice of any changes to the syllabus through myCourses/email.

Concluding statement

Most importantly, please be assured that I want students to learn and to receive the good grades they deserve. So please make an appointment with me should you have undue difficulty with your work in the course.

Bibliography

- [1] Bittman, T. J., Dawson, P., Margevicius, A., & Weiss, G. J. Magic Quadrant for x86 Server Virtualization Infrastructure. June 2013. ID: G00251472.
- [2] Daniels, J. Server virtualization architecture and implementation. In Magazine Crossroads, Volume 16 Issue 1. September 2009. 8-12.
- [3] Atwal, R., Tian, U., & Wu, J. Forecast: x86 Server Virtualization, Worldwide, 3Q12 Update. October 2012. ID: G00229266.
- [4] Bittman, T. J., Dawson, P., Margevicius, A., & Weiss, G. J. Magic Quadrant for x86 Server Virtualization Infrastructure. June 2013, ID: G00251472.
- [5] Creasy, R. J. The Origin of the VM/370 Time-sharing System. IBM J. RES. DEVELOP. VOL. 25 NO. 5. SEPTEMBER 1981.
- [6] Portnoy, M. Virtualization Essentials. April 2012, ISBN: 978-1-118-17671-9, Sybex.
- [7] Intel, "Moore's Law Technology",
<http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>
- [8] Carroll, M., Kotzé, P., & Merwe, A. Going virtual: popular trend or real prospect for enterprise information systems. 12th International Conference on Enterprise Information Systems, Funchal, Madeira, Portugal, 8- 12 June 2010, pp 214-222.
- [9] Redhat, KVM Kernel Based Virtual Machine.
- [10] Anderson, A.V., Bennett, S.M., Kagi, A., Leung, F.H., Martins, F.C.M., Neiger, G., Rodgers, D., Santoni, A.L., Smith, L., & Uhlig, R. Intel Virtualization Technology. In Computer, Volume 38 Issue 5. May 2005. 48-56.
- [11] VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assist.
- [12] Campbell, S., & Jeronimo, M. An Introduction to Virtualization.
- [13] National University of Singapore, "Virtual Memory",
<http://www.comp.nus.edu.sg/~lubomir/PROOFS/ch8.pdf>
- [14] Wikipedia, "Translation lookaside buffer",
http://en.wikipedia.org/wiki/Translation_lookaside_buffer

- [15] AMD, AMD-V Nested Paging. July 2008.
- [16] Carvalho, H. E. T., Duarte O. C. M. B, Ferraz, L. H. G., & Pisa, P. S. New I/O Virtualization Techniques. Federal University of Rio de Janeiro, GTA/COPPE - Rio de Janeiro, Brazil.
- [17] Intel, Intel Virtualization Technology for Directed I/O (VT-d): Enhancing Intel platforms for efficient virtualization of I/O devices.
- [18] Lowe, S. "What is SR-IOV?" December 2009.
- [19] Intel, An Introduction to SR-IOV Technology. January 2011.
- [20] Lowe, S., & Marshall, N. Mastering VMware vSphere 5.5. October 2013, ISBN: 978-1-118-66114-7, Sybex.
- [21] VMware, The Architecture of VMware ESXi.
- [22] VMware, Understanding Memory Resource Management in VMware ESX Server.
- [23] Finn, A., Flynn, D., Lownds, P., & Luescher, M. Windows Server 2012 Hyper-V installation and Configuration Guide. March 2013, ISBN: 978-1-118-48649-8, Sybex.
- [24] An Overview of the Hyper-V Architecture,
http://www.virtuatopia.com/index.php/An_Overview_of_the_Hyper-V_Architecture
- [25] Microsoft, Hyper-V Dynamic Memory Overview,
<http://technet.microsoft.com/en-us/library/hh831766.aspx>
- [26] Border, C., Gonzalez, C., & Oh, T. Teaching System Administration in Amazon's Elastic Cloud Computing. In Proceedings of the 14th annual ACM SIGITE conference on Information technology education (SIGITE 12). October 2013. 149-150.
- [27] Adams, K., & Agesen, O. A comparison of software and hardware techniques for x86 virtualization. In Newsletter ACM SIGPLAN Notices - Proceedings of the 2006 ASPLOS Conference, Volume 41 Issue 11. November 2006. 2-13.
- [28] Bugnion, E., Devine, S., Rosenblum, M., Sugerman, J., & Wang, E. Y. Bringing Virtualization to the x86 Architecture with the Original VMware Workstation. In Journal ACM Transactions on Computer Systems (TOCS), Volume 30 Issue 4. November 2012. Article No.12.

- [29] Lath, R., Mohapatra, S., & Sahoo, J. Virtualization: A Survey On Concepts, Taxonomy And Associated Security Issues. In Computer and Network Technology (ICCNT), 2010 Second International Conference on. April 2012. 222-226.
- [30] Goldberg, R. P., & Popek, G. J. Formal requirements for virtualizable third generation architectures. In Magazine Communications of the ACM, Volume 17 issue 7. July 1974. 412-421.
- [31] Waldspurger, C. A. Memory Resource Management in VMware ESX Server. In Newsletter ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, Volume 36 Issue SI. December 2002. 181-194.
- [32] Le, C. H. H. Protecting Xen hypercalls. The University of British Columbia. July 2009.
- [33] Mack, C.A. Moore's Law 3.0. In Microelectronics and Electron Devices (WMED), 2013 IEEE Workshop on. April 2013. Xiii.
- [34] Li, P., Noles, J., & Toderick, L. Provisioning Virtualized Datacenters through Virtual Computing Lab. In Frontiers in Education Conference (FIE), 2010 IEEE. October 2010. T3C-1 – T3C-6.
- [35] Warrilow, M. Report Highlight for Market Trends: x86 Server Virtualization, Worldwide, 2013. February 2014. ID: G00262332.
- [36] Qiu, O. Z., & Yue, Z. Research on Application of Virtualization in Network Technology Course. In Computer Science & Education (ICCSE), 2012 7th International Conference on. July 2012. 357-359.
- [37] Dampier, D.A, Dandass, Y.S., & Shannon, S.T. Teaching Hypervisor Design, Implementation, and Control to Undergraduate Computer Science and Computer Engineering Students. In System Science (HICSS), 2012 45th Hawaii International Conference on. January 2012. 5613-5622.
- [38] Amin, M.N, Dey, P.P., Romney, G.W., & Sinha, B.R. The agility, flexibility and efficiency of hypervisors in engineering education. In Information Technology Based Higher Education and Training (ITHET), 2013 International Conference on. October 2013. 1-8.
- [39] Caminero, A.C., Hernandez, R., Pastor, R., Robles-Gomez, A., & Ros, S. Using Virtualization and Automatic Evaluation: Adapting Network Services Management Courses to the EHEA. In Education, IEEE Transactions on Volume 55 Issue 2. May 2012. 196-202.

- [40] Dobrilović, D., & Stojanov, Z. Using Virtualization Software in Operating Systems Course. In Information Technology: Research and Education, 2006. ITRE '06. International Conference on. October 2006. 222-226.
- [41] Weber, S. Using Virtualization Technology to Teach Operating System Concepts : tutorial presentation. In Journal of Computing Sciences in Colleges, Volume 23 Issue 6. June 2008. 72-72.
- [42] Armitage, W., Daniels, T., Gaspar, A., Langevin, S., & Sekar, R. The Role of Virtualization in Computing Education. In Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08). March 2008. 131-132.
- [43] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, L., & Warfield, A. Xen and the Art of Virtualization. In Proceedings of the nineteenth ACM symposium on operating systems principles (SOSP '03). December 2003. 164-177.
- [44] Border, C. The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes. In Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE '07). March 2007. 576-580.
- [45] Huang, D., Tsai, W., & Xu, L. V-Lab: A Cloud-based Virtual Laboratory Platform for Hands-On Networking Courses. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12). July 2012. 256-261.

Tables of Figures

Figure 1. Magic Quadrant for x86 Server Virtualization Infrastructure

Figure 2. IBM System/370 Model 158

Figure 3. History and evolution of virtualization

Figure 4. Moore's Law: transistor count

Figure 5. A sample virtual machine monitor

Figure 6. Privilege levels for x86 platform

Figure 7. Ring deprivileging

Figure 8. Full Virtualization Using Binary Translation

Figure 9. Paravirtualization

Figure 10. Hardware Assisted Virtualization

Figure 11. Type 1 (Bare-metal) hypervisor

Figure 12. Type 2 (Hosted) hypervisor

Figure 13. Address/Memory Translation

Figure 14. Software Based Sharing

Figure 15. Direct Assignment (direct I/O device pass-through)

Figure 16. Single Root I/O Virtualization

Figure 17. Magic Quadrant for x86 Server Virtualization Infrastructure (2011)

Figure 18. Magic Quadrant for x86 Server Virtualization Infrastructure (2012)

Figure 19. Magic Quadrant for x86 Server Virtualization Infrastructure (2013)

Figure 20. VMware ESXi Architecture

Figure 21. Transparent Page Sharing (TPS)

Figure 22. Ballooning

Figure 23. Hyper-V Architecture

Figure 24. Increasing minimum memory error

Figure 25. Lab setup for VMware vSphere lectures

Figure 26. Lab setup for Microsoft Hyper-V lectures