IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 6, June 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

Virtualization Technology

Jamil Abdul Kareem Almuttawa

ABSTRACT

Virtualization can afford various benefits, great efficiency in CPU utilization, green IT environment with low power consumption, central environment control, more availability, reduces project timelines by eliminating hardware procurement, improves disaster recovery capabilities, more central control of the desktop, and improve outsourcing services.

INTRODUCTION

Nowadays, Virtualization gained popularity in many different areas such as server consolidation, cloud computing and information security. This is greatly due to an increase in hardware performance in the past decade and the goal to reduce capital and operational costs within the data center. Virtualization can help IT managers fully maximize the capabilities of their hardware investment by increasing the number of operating systems and applications running on a single physical server.

Several projects exist that make use of virtualization in one form or another to provide a virtual lab environment of varying levels of sophistication and scalability to students. The primary goal for all of them; to reduce the physical hardware necessary to administrate and maintain computer security lab assignments. When developing such a platform for academia several factors become must also be considered. Things such as the hardware requirement, guest operating systems, type of assignments, and how it is to be administrated and accessed become very important. In this day in age, the method of providing virtualization is key. There are a multitude of platforms and options to consider each having their own advantages.

VIRTUALIZATION DEFINITION

Virtualization is a methodology of dividing the resources of a computer into multiple execution environments, by applying one or more technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.

Virtualization is an abstraction layer meaning it abstracts the operating system from the hardware so with a typical physical server you install the operating system right on top of hardware and it has drivers that talk to the hardware well it does not work that way with VMware. You create the virtualization layer there with VMware vsphere or Citrix xenserver or Microsoft Hyper-V, whatever it is, if it is a type 1 hypervisor which means it installs directly on the hardware you create this additional layer between the hardware of the operating system and then you have got these multiple containers that contain your virtual machines. So, the operating system is actually running inside each virtual machine and the operating system has virtual devices so those virtual devices go through the VMware layer or the hypervisor layer to talk to the physical devices so this is what virtualization is in general.

Virtualization is considered a method of logically dividing mainframes to allow multiple applications to run simultaneously. Before that, systems could only execute a single application at a time. According to the high cost of these mainframe systems, many organizations found it hard to invest in a system, which could only due one thing at a time. The advent of the VM mainframe from IBM and has continued to evolve since.

With the adoption of Windows and Linux operating systems lead to a shift away from mainframe based applications to the more distributed client-server based applications. It was now common place for these systems to run multiple applications simultaneously although not in the sense that it was done in the mainframes. It wasn't until the early 2000s where virtualization began to evolve as x86 based systems became more powerful.

As corporate data centers began to grow so did the cost of supporting the high number of systems. Especially as applications were generally dedicated their own server to avoid conflicts with other applications. This practice caused a waste in computing resources as the average utilization for many systems was only 10% to 15% of their possible capacity. It is at this point many companies started looking at virtualization for a solution. Unfortunately, the now common x86 systems were not designed with virtualization support in mind, which caused challenges when attempting to handle some of the privileged instructions in the x86 architecture.

VIRTUALIZATION CHARACTERISTICS

To meet the requirements of a dynamic business, your IT organization needs to deliver the highest levels of application performance and availability in a cost-effective manner. Here are some characteristics that application should have in order to achieve virtualization in the right way.

1. Proven Technology

When you deploy a virtualization solution, you want to have the confidence that your environment is based on proven technology, is widely used, and is widely supported by application vendors.

2. Integrated Management

The virtualization market is filled with management tools that were built for physical environments and later reworked to manage virtual systems. That is not the case with solutions from VM, which were purpose built to manage dynamic virtual and cloud environments.

3. Reliability

When your business operations depend on your IT services, you want to have the confidence that comes with a reliable, predictable computing environment designed for virtualization.

4. High Availability

VT with Operations Management is designed to deliver enhanced availability and performance for both your business-critical and next-generation applications. A rich feature set is focused on making sure applications deliver the best possible performance based on your policies and service-level agreements.

5. Disaster Recovery

When your business depends on your applications, you need to have tools in place that allow you to recover data quickly in times that disrupt your IT systems. VT allows this feature in a manner of software packages are ready to be run at any time.

VIRTUALIZATION TYPES

Companies of all sizes are embracing virtualization as a way to cut IT expenses, enhance security, and increase operational efficiency. While the benefits of virtualization are self-evident, many people are still in the dark when it comes to the many different types of virtualization. Here, we will show you some of the most common virtualization methods and why they are valuable for your business.

Application Virtualization

The encapsulation of an application into a self-contained distributable package which is isolated from the underlying operating system and other applications.

There are currently three types of application virtualization technologies common to products available today:

Redirection: an application is placed into a cache folder and registry location when executed the client falls the applications to think it is being executed from standard locations.

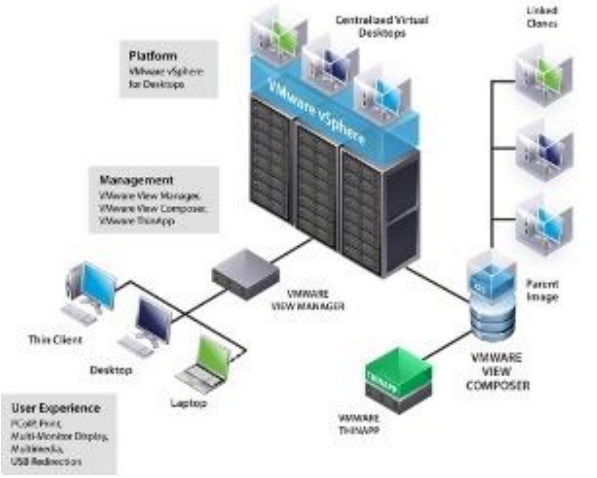
Layering: applications are stored in virtual discs with the client overlays with a system disc to provide a merged view.

Virtual environment: each application is stored in a micro virtual environment containing its own virtual file system and registry subsystem the application is executed within the virtual environment which has a merged view with the local system.

This process where applications get virtualized and are delivered from a server to the end user's device, such as laptops, smartphones, and tablets. Therefore, instead of logging into their computers at work, users will be able to gain access to the application from virtually anywhere, provided an Internet connection is available. This type of virtualization is particularly popular for businesses that require the use of their applications on the go.

Desktop Virtualization

Similar to Application Virtualization mentioned above, desktop virtualization separates the desktop environment from the physical device and configured as a "virtual desktop infrastructure" (VDI). The major advantages of desktop virtualization is that users are able to access all their personal files and applications from any location and on any PC, meaning they can work from anywhere without the need to bring their work computer. It also lowers the cost of licensing for installing software on desktops and maintenance and patch management is very simple, since all of the virtual desktops are hosted at the same location.



You chose VMware view as your desktop virtualization solution it uses VMware vSphere as the hypervisor so you see the three racks of servers that you got a bunch of servers on top of those servers you loaded VMware vSphere as the hypervisor and then one of the servers is your VMware view manager. So, this is also a connection broker this is where your end-users devices whether they are thin clients' desktops or a local mode laptop computers this is where they connect. They connect to that connection broker or the VMware view manager. It is going to direct them to their virtual machine desktop that is running on top of VMware vSphere. So, as a manager you would use the VMware view manager to administer the virtual desktop infrastructure and the end-users are connecting through the view manager to their virtual machine desktop no matter where they are, if they are out on the internet, if they are at another company site, if they are on their mobile device or if they are sitting in their cubicle at the corporate headquarters. No matter where they are they are going through the connection broker to get to their virtual desktop. That desktop has all their company applications. It has access to their user data and it is all stored in the data center just like all the other enterprise-grade applications.

Hardware Virtualization

This is perhaps the most common type of virtualization today. Hardware virtualization is made possible by a virtual machine manager (VM) called the "hypervisor". The hypervisor creates virtual versions of computers and operating systems and consolidates them into one large physical server, so that all the hardware resources can be utilized more efficiently. It also enables users to run different operating systems on the same machine at the same time.

Network Virtualization

Network virtualization is a method that combines all physical networking equipment into a single resource. It is the process of dividing bandwidth into multiple, independent channels, each of which can be assigned to servers and devices in real time. Businesses that would benefit from network virtualization are ones that have a large number of users and need to keep their systems up and running at all times. With the distributed channels, your network speed will increase dramatically, allowing you to deliver services and applications faster than ever before.

Virtualization is composed of two technologies NFV and SDN. Virtualizing the network consists of separating software from hardware and network equipment so that network functionalities become independent of the physical equipment supporting them as all functionalities are found in software and mounting the physical machines the same server can be used for several purposes depending on the software installed on it as functionalities reside exclusively in software it is possible to package each network function in one or more virtual machines and to decide where to execute them. This is known as NFV (Network Function Virtualization). This new concept is now feasible because a set of advances in general-purpose servers has managed to increase by an order of magnitude the performance that software can achieve in the handling of data.

Hardware resources must be allocated with great care in order to achieve high and predictable performance so that virtualized network functions can work at the same speed as traditional network functions. At the same time those network functions must be interconnected with each other in a coherent way in order to provide network services. These interconnections can be managed from a centralized point, the centralization of the control panel is what is known as SDN (Software-Defined networks).

Storage Virtualization

This type of virtualization is very easy and cost-effective to implement, since it involves compiling your physical hard drives into a single cluster. Storage virtualization is handy when it comes to planning for disaster recovery, since the data stored on your virtual storage can be replicated and transferred to another location. By consolidating your storage into a centralized system, you can eliminate the hassles and costs of managing multiple storage devices.

VIRTUALIZATION DEPOLYMENT MODELS

Virtualization exploits the similarity in architectures of the guest and host platforms to cut down the interpretation latency. Most of the today's world's commercial PC emulators use this virtualization technique on popular x86 platforms to make it efficient and its use, viable and practical. Virtualization technique helps map the virtual resources to physical resources and use the native hardware for computations in the virtual machine. When the emulated machine needs to talk to critical physical resources, the simulator takes over and multiplexes appropriately.

For such a virtualization technology to work correctly, the VM must be able to trap every privileged instruction execution and pass it to the underlying VMM to be taken care of. This is because, in a VMM environment, multiple VMs may each have an OS running that wants to issue privileged instructions and get the CPU's attention. When a trap occurs during privileged instruction execution, rather than generating an exception and crashing, the instruction is sent to the VMM. This allows the VMM to take complete control of the machine and keep each VM isolated. The VMM then either executes the instruction on the processor, or emulates the results and returns them to the VM. However, the most popular platform, x86, is not fully virtualizable, i.e. certain supervisor (privileged) instructions fail *silently* rather than causing a convenient trap when executed with insufficient privileges. Thus, the virtualization technique must have some workaround to pass control to the VMM when a faulting instruction executes. Most commercial emulators use techniques like *code scanning* and *dynamic instruction rewriting* to overcome such issues. In this section, we shall explore the techniques used by a number of commercial PC emulators to create correct and efficient virtualized machines and some of their features and shortcomings.

- **VMWare**

VMware is a virtual machine company with three levels of VM products: VMware Workstation, VMware GSX Server, and VMware ESX server. VMware's VMMs can be *standalone* or *hosted.* A standalone VMM is basically a software layer on the base hardware that lets users create one or more VMs. These are similar to operating systems; require device drivers for each hardware device, and are typically limited in hardware support. Such VMMs are typically used in servers, VMware ESX server being a prime example of such an architecture. A hosted VMM runs as an application on an existing host operating system. It can take advantage of the host operating system for memory management, processor scheduling, hardware drivers, and resource management. VMware Workstation group of products use this hosted virtual machine architecture.

VMware Workstation has three components: the *VMX driver* and *VMM* and the *VMware application*. The VMX driver is installed within the operating system to gain the high privilege levels required by the virtual machine monitor. When executed, the VMApp loads the VMM into kernel memory with the help of VMX driver, giving it the highest privilege. The host OS, at this point, knows about the VMX driver and the VMApp, but does not know about the VMM. The machine now has two worlds: the *host world* and the *VMM world*. The VMM world can communicate directly with the processor hardware or through the VMX driver to the host world. However, every switch to the host world would require all the hardware states to be saved and restored on return, which makes switching hit the performance. When the guest OS or any of its applications run purely computational programs, they are executed directly through the VMM in the CPU. I/O instructions, being privileged ones, are trapped by the VMM and are executed in the host world by a world switch. The I/O operations requested in the VM are translated to high-level I/O related calls and are invoked through the VM App in the host world, and the results are communicated back to the VMM world. This makes the overall VM run slow for I/O intensive applications.

ESX server product is installed on a bare machine without any operating system. It gives a console interface to create and configure VMs. The product typically finds use in server consolidation and web hosting. Since there is no host operating system, VMM has to handle all the I/O instructions, which necessitates the installation of all the hardware drivers and related software. It implements shadow versions of system structures such as page tables and maintains consistency with the virtual tables by trapping every instruction that attempts to update these structures. Thus, there exists one extra level of mapping in the page table. The virtual pages are mapped to physical pages through the guest operating system's page table. The physical page (often called *frame*) is then translated to the machine page by the VMM, which eventually is the correct page in physical memory. This helps the ESX server better manage the overall memory and improve the overall system performance. It uses various other techniques to increase the overall efficiency, and level of isolation to keep each VM independent from another, making it a reliable system for commercial deployment.

The newer versions of VMware Workstation come with some of the striking features. *Pointer integration* with the host desktop allows the use to move the mouse pointer seamlessly in and out of the VMware Application's display window like it happens with any other window-based application. *File sharing* allows the user to share files and folders between the host and the guest machines to help easy transfer of data from and to the virtual machines for backing up and other purposes. *Dynamic display resizing* lets the user dynamically resize the VMware Application's display window like any other window. This is not so trivial realizing the fact that every resize operation changes the screen resolution for the virtual machine.

VMware Workstation supports a variety of networking setups to help user connect to the network according to his/her convenience. It has the provision of a virtual ethernet hub that connects all the virtual ethernet adapters that exist in the VMs to create a LAN within the host computer. It also supports bridged networking through the external ethernet card connected to the real network outside the VM. Network address translation (NAT) is also supported. Among other features, it passes the host USB devices to the virtual machines that lets user connect and work with any USB device inside of a VM. However, VMware limits the maximum memory to be allocated across all the active VMs to be 4GB. Although at this time it does not really look like a limitation, it is for the world to see how things develop in the next few years.

- **Virtual PC**

Microsoft's Virtual PC, recently acquired from Connectix, is a product very similar to what is offered by VMware Workstation. It is based on the Virtual Machine Monitor (VMM) architecture and lets the user create and configure one or more virtual machines. Apart from the features supported by VMware, it provides two distinguishing functionalities. It maintains an *undo disk* that lets the user easily undo some previous operations on the hard disks of a VM. This enables easy data recovery and might come handy in several circumstances. The other striking feature is *binary translation*, which it uses to provide x86 machines on Macintosh-based machines.

There are a number of shortcomings that the Virtual PC possess in terms of features when compared to VMware. Linux, FreeBSD, OpenBSD, Solaris, etc are not supported as guest OSes in Virtual PC. The Virtual PC VMs do not have support for SCSI devices, unlike VMware workstation, although some SCSI disks are recognized as IDEs by the VMs. It does not let user add or upgrade the hardware set for a VM. Once configured, it makes it impossible to change the hardware devices a VM possesses later on. Linux or other exotic operating systems are not available as host OS.

- **Denali**

Although virtual machines provided by the likes of VMware Workstation and Microsoft Virtual PC are very efficient and practical to use supporting almost all the PC-like features with all the ease, due to design limitations it is difficult to create and use thousands of them simultaneously. The way virtualization is achieved in the VMM makes it difficult to scale it to high numbers. For example, the interrupt handling mechanism is hard to scale beyond a few active VMs, if multiplexed simultaneously. The same is the case for memory management, world switching, and so on. However, there might be legitimate reasons to have large numbers of active virtual machines for various purposes. The University of Washington's Denali project tries to address this issue and come up with a new virtualization architecture to support thousands of simultaneous machines, which they call Lightweight Virtual Machines. Using a technique, called *paravirtualization*, it tries to increase the scalability and performance of the Virtual Machines without too much of implementation complexity.

The *paravirtualization* technique modifies the traditional virtualization architecture for new customized guest operating systems (unlike VMware Workstation, that supports legacy OSes) to obtain extra performance, and high scalability. This new architecture comes up with new interfaces for the customized guest operating systems. The paravirtualized architecture provides modified architectural features that makes the implementation of guest OSes simple yet versatile. *Virtual instructions*, equivalent to system calls in traditional architecture, expose rich and simple instructions for the upper layer by grouping and optimizing commonly used instructions. The new architecture exposes a set of *virtual registers* for ease of data transfer between the virtualization layer and the virtual machines. It also provides a simplified architectural interface to be exported by the virtual I/O devices. Among other things, it supports a modified interrupt delivery, and does not support the virtual memory concept. All these modifications are incorporated aiming at a simpler implementation with low overhead to make the overall system scalable and the VMs lighter.

- **Xen**

The discussions so far have been concentrating on full virtualization, where applications and the operating system within a VM live in a complete virtual world with no knowledge of the real machine whatsoever. Although many a times this has been the goal of virtualization, there may be cases where an application or the operating system running within a VM might desire to see both the real as well as virtual resources and use the information to its benefit. For example, seeing both real and virtual time might help the guest OS better support time-sensitive tasks and come up with good round trip time (RTT) estimates for handling TCP timeouts. Likewise, seeing real machine addresses might help improve performance by using super pages and page coloring. This apart, a full virtualization is always tricky and cumbersome when implemented on x86 due to its inherent problem of not being a virtualizable architecture. X86, being an uncooperative machine architecture, makes the task of achieving high performance with a strong resource isolation in virtualization very difficult. In addition, completely hiding the effects of resource virtualization from guest OSes risks both correctness and performance.

All these along with issues like QoS, security, and denial of service motivate the researchers in University of Cambridge come up with a modified architecture for virtualization, called Xen. Xen exports a paravirtualized architecture in each of its VMs to maximize performance and resource isolation yet maintaining the same application binary interface (ABI) as commodity operating systems. Although it does require the operating systems to be ported, the porting effort is kept as low as possible. It aims at supporting around a hundred VM instances within a single physical machine within a reasonable performance hit. Although Denali uses a paravirtualized architecture for more or less the same purposes, they both have diffrent targets. Denali is designed to supposed thousands of virtual machines running network services, the vast majority of which are small-scale and unpopular. Since the applications used in Denali are customized ones, the paravirtualization architecture does not have to guarantee the *ABI compatibility* and thus can elide certain architectural features from its VM interface. No support for x86 segmentation is one such example although ABIs in most of the OSes including Windows XP, Linux, and NetBSD export this feature. Denali VMs are designed with the aim of hosting a *single application, single-user unprotected guest OS* (e.g. Ilwaco) and thus does not have support for virtual memory which is a common feature in almost all the modern OSes. Denali VMM performs all the paging work to and from the disks that are vulnerable to thrashing should there be any malicious VM, while Xen relies on the guest OS to do all the paging. Denali virtualizes *namespaces*, whereas Xen believes *”secure access control”* within the hypervisor is sufficient to ensure protection while making physical resources directly accessible to guest OSes .The para virtualization exports a new virtual machine interface that aims at improving the performance and scalability as compared to the other commercial VMMs. A new *lightweight event mechanism* replaces the traditional hardware interrupts in the x86 architecture for both CPU as well as the device I/O. This greatly improves the performance and scales to great numbers. *Asynchronous I/O rings* are used for simple and efficient data transfers between the VMs and the hypervisor (Xen’s VMM or simply Xen, in short). For security purposes, *descriptor tables* for exception handlers are registered with Xen by each of the VMs and aside the page faults, the handlers remain the same. Guest OS may install ’fast’ handler for system calls, allowing direct calls from an application into its guest OS avoiding the indirection through Xen on every call. For efficient page table and TLB management, Xen exists in a 64MB section of every address space. This avoids a TLB flush when entering and leaving the hypervisor. However, this also means a restricted segmentation that disallows installation of fully-privileged segment descriptors and the top end of the linear address space can not be overlapped. Guest OSes have the direct access to hardware page tables, however, updates are batched and validated by Xen. This allows Xen to implement a secure but efficient memory management technique when compared to VMware where every update to the page table is trapped by VMM and updated. In particular, when a new process is created through fork, the number of updates is enormous which might hit the performance badly in VMware. Using batched updates, as in Xen, helps it a lot. Each guest OS is provided a timer interface and is aware of both ’real’ and ’virtual’ time. In this way, even though the hypervisor exports a modified VM interface as compared to a traditional x86 architecture, it tries to build a more robust architecture that preserves all the features that are of importance to application binaries yet keeping the porting effort for the guest OSes minimal.

- **Plex86**

Plex86 project works toward an open-source x86 simulator with virtualization. It uses the virtualization technique to improve the efficiency of a virtual machine such as Bochs. Virtualization is a technique that is used to take advantage of the hardware similarity of the guest and the host machine to allow large portions of the simulation to take place at the native hardware speed. When the simulated machine talks to the hardware, or enters certain privileged modes (such as the ”kernel mode”), the simulator takes control and simulates the code in software at a much slower speed, as the Bochs does. Thus, virtualization helps Plex86 run much faster compared to Bochs, however, the portability is lost. Another version of Plex86 is being developed just to create a partial virtual machine that is able to support Linux and which runs much faster than the full scale virtual machine.

- **User-mode Linux**

User-mode Linux, or UML, is an open source project that lets the user run Linux on top of Linux. Basically, it gives a virtual machine on which a Linux version can execute as it does on a physical machine, and everything implemented in the user-level. Unlike previous ones that use the VMM right on the base hardware, this uses a different implementation being on top of the operating system and in the user-space. Implementation aside, the abstraction level remains more or less similar to the previous ones. It lets the user configure virtual hardware resources that would be available for the guest Linux kernel. Since everything runs in the user-level, safety is assured. It's hardware support comes in the form of virtual devices that make use of the physical resources. Among the devices supported are, block devices, consoles, serial lines, network devices, SCSI devices, USB, Sound, and so on. The UML runs its own scheduler independent of the host scheduler, runs is own virtual memory system, and basically supports anything that is not hardware specific. It also supports SMP and highmem. The virtual console driver implementation lets the user attach it to a number of interfaces available on the host: file descriptors, ptys, ttys, pts devices, and xterms.

Implementation of UML involves a port of the Linux kernel to the Linux system call interface rather than to a hardware interface. In this regard, the virtual machine and the guest Linux kernel are tightly coupled. Executing totally in the user space, the major challenge it faces is to be able to intercept the system calls in the virtual kernel, as they would naturally go to the real host kernel. Using the Linux ptrace facility to track system calls, it diverts the system calls made by processes running within the Virtual Machine to the user space kernel to execute them. Similarly, traps are implemented through Linux signals. Kernel and the processes within the VM share the same address space; and conflicts with process memory are avoided by placing the kernel text and data in areas that processes are not likely to use. Each process in the virtual machine gets its process in the host kernel. In order for the virtual kernel's data to be shared across all the processes in the VM, its data segment is copied into a file, and the file is mapped shared to all the processes. Using such tricks, it implements, that too within a reasonable overhead, the user-space virtual machine.

- **Cooperative Linux**

CoLinux, as it is often called, is a variation of User-mode Linux. It is a port of the Linux kernel that allows it to run as an unprivileged lightweight virtual machine in kernel mode, on top of another OS kernel.

It allows Linux to run under any OS that supports loading drivers, such as Windows, with some minor porting effort.

Cooperative Linux works in parallel with the host kernel. In such a setup, each kernel has its own complete CPU context and address spec, and decides when to give the control back to its partner. However, only one of the two kernels has the control on physical hardware (the host kernel), and the other (guest kernel) is provided only with virtual hardware abstraction. The only requirement on the host kernel is that it should allow to load the colinux portable driver to run in ring 0 and allocate memory. The colinux VM uses only one host process, called the Super Process, for itself and its processes. It uses the portable driver to switch the context to the host kernel and back as well as to load the kernel from a file during startup. Using a forwarding technique, colinux handles the interrupts by making use of its own code and the context switches appropriately. The overall performance is comparable to UML.

VIRTUALIZATION PROS

Today's IT intensive enterprise must always be on the lookout for the latest technologies that allow businesses to run with fewer resources while providing the infrastructure to meet today and future customer needs..

Server Consolidation

It is not unusual to achieve 10:1 virtual to physical machine consolidation. This means that ten server applications can be run on a single machine that had required as many physical computers to provide the unique operating system and technical specification environments in order to operate. Server utilization is optimized and legacy software can maintain old OS configurations while new applications are running in VMs with updated platforms.

Although a server supporting many VMs will probably have more memory, CPUs, and other hardware it will use little or no more power and occupy the same physical space reducing utilities costs and real estate expenditures.

Testing and development

Use of a VM enables rapid deployment by isolating the application in a known and controlled environment. Unknown factors such as mixed libraries caused by numerous installs can be eliminated. Severe crashes that required hours of reinstallation now take moments by simply copying a virtual image.

Dynamic Load Balancing and Disaster Recovery

As server workloads vary, virtualization provides the ability for virtual machines that are over utilizing the resources of a server to be moved to underutilized servers. This dynamic load balancing creates efficient utilization of server resources.

Disaster recovery is a critical component for IT, as system crashes can create huge economic losses. Virtualization technology enables a virtual image on a machine to be instantly re-imaged on another server if a machine failure occurs.

Virtual Desktops

Multinational flexibility provides seamless transitions between different operating systems on a single machine reducing desktop footprint and hardware expenditure.

"…Parallels Desktop for Mac, a virtual machine application. Instead of Boot Camp's dual-boot approach, Parallels Desktop runs Windows XP directly on the Mac OS desktop (in what Parallels calls "near-native performance")--allowing you to run both OSs simultaneously and switch back and forth seamlessly." Daniel A. Begun, CNet: Heresy: Windows XP performance on a Mac.

Improved System Reliability and Security
Virtualization of systems helps prevent system crashes due to memory corruption caused by software like device drivers. VT-d for Directed I/O Architecture provides methods to better control system devices by defining the architecture for DMA and interrupt remapping to ensure improved isolation of I/O resources for greater reliability, security, and availability.

VIRTUALIZATION CONS

Cost
The upfront cost can be much higher and, depending on how high of an availability you want, you will need to be willing to design the system for your needs now and in the future.

Complexity
If you are not familiar with the hardware and network aspects of the whole setup, it can be a daunting task to figure out. Routing rules and VLAN's continue to add complexity, especially if security is a concern.

Often the hardware is bundled together
In one location making a single disaster more likely to cause significant down time. However, there are ways around this.

Hardware keys
Yes, you can use hardware keys. You can bind a USB port to a specific virtual machine. However, you are not able to move the virtual machine without physically moving the key as well.

Add-on hardware
In the past, you were not able to add on older PCI hardware and share it with the virtual machine. This has changed, but it does not work 100% of the time. I would recommend testing it thoroughly before deploying. Of course, this also limits which machine a virtual machine can run on because it will need to be bound to that piece of hardware.

CONCLUSION

This paper presented essential terms related to virtualization with the aim to answer questions frequently asked by people who are in the IT field. These terms included its Introduction, definition, characteristics, types, deployment models and finally pros and cons.

REFERENCES:

1. *Christine Leja. (2016, January 12). AITP research: "Virtualization and its benefits" - Association of Information Technology Professionals.*
2. *Marshall, D. (2011, November 2). Top 10 benefits of server virtualization.*
3. *5 different types of virtualization. (2016, January 29).*
4. *"vmw-5-essential-characteristics-ebook.pdf"*
5. *Thomas Burger. (2012, March 5). The advantages of using Virtualization technology in the enterprise.*
6. *"Virtualization in education" (PDF). IBM. October 2007. Retrieved 6 July 2010. A virtual computer is a logical representation of a computer in software. By decoupling the physical hardware from the operating system, virtualization provides more operational flexibility and increases the utilization rate of the underlying physical hardware.*
7. *Nanda, S., Chiueh. T. A survey on virtualization technologies 2005, http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf*
8. *David Ward, & Cross Company. (2015, March 16). The pros and cons of Virtualizing a control system. Retrieved from http://innovativecontrols.com/blog/pros-and-cons-virtualizing-control-system*