Course Title: Operating System

Course code: CSC 3401

Topic: Virtualization & the cloud

Group Members

Student Name	Matric No.
Abid Ebna Saif Utsha	1433527
Mahfuzealahi Noman	1515803
Tahmina Rahman	1221638
Raha Binte Rashid	1527434

Section : 02

Lecturer : Dr. Ali A. Alwan

Date of Submission: 18/05/2018

Table of Contents:

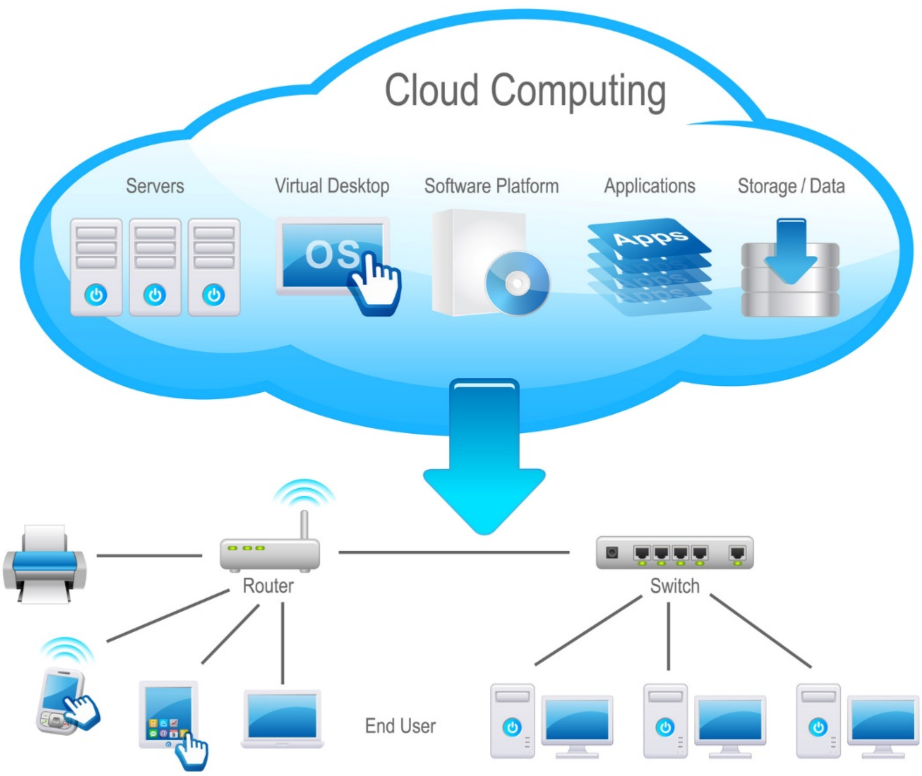
1.0 Abstract

Virtualization and cloud computing are one of the most important driving factors in the development of computer science industry. Cloud computing makes effective use of distributed environments for tracking large scale computation problems on vast data set. Virtualization is the key enabler of cloud computing. It provides the capability of sharing the server clusters as a pool of computing resources and the capability of dynamically mapping virtual resource to customers and applications. Many existing cloud platforms used virtualization to provide isolated compute resources. This paper will review the basic knowledge about virtualization and the cloud.

2.0 Introduction

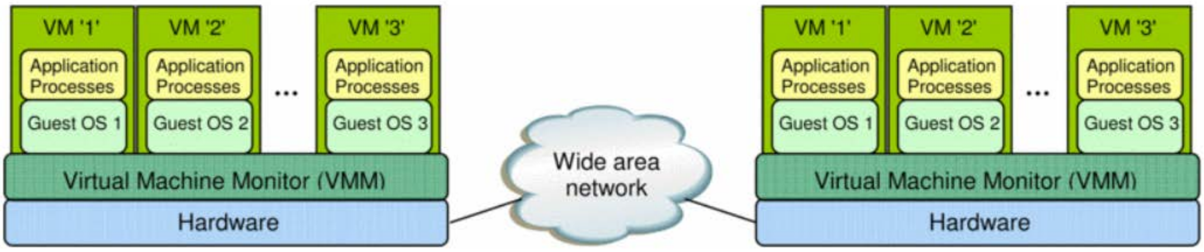
In computing, virtualization refers to the act of creating a virtual version of something, including virtual computer hardware platforms, storage devices and computer network resources. For example – a computer has many physical devices like- CPU, memory, secondary disk drives etc. On average consume, people are not using even ten percent on their computer potentials. Virtualization helps to better utilize the computer's resource.

Cloud computing really is accessing resources and services needed to platform functions with dynamically changing needs, the cloud is a virtualization of resources that maintains and manages itself.



For simplicity, virtualization software makes multiple operating system and multiple application to run on the same server at the same time. Sometimes, most of the people got confused about virtualization and cloud computing. But both are different because virtualization is software that manipulates hardware, while cloud computing refers to a service that results from that manipulation. Virtualization is a fundamental element of cloud computing and helps deliver on the value of cloud computing.

The cloud most often includes virtualization products to deliver the compute service, the difference is true cloud provides self-service capability that is not inherent in virtualization.



3.0 History

Although server virtualization technology is currently receiving much attention, the concept is not new. Actually, the idea came in the mid-1960s, when the Giants and expensive computers of the day reached to a high processing speed, but they were unable to seize the expensive computing time due to management processes that needed to be done manually by the operator. To get the best out of expensive computer processing, it was necessary to run multiple processes in parallel. Thus, arose the concept of time sharing, which culminated with the idea of virtualization.

This concept means sharing time, i.e., idle time between the processes are shared with other processes to streamline the system. Multiple jobs are executed simultaneously, and the CPU meets each job for a little while, one by one in sequence. The time dedicated to each job are small enough so that users can interact with each program to recognize that there are no other programs running.

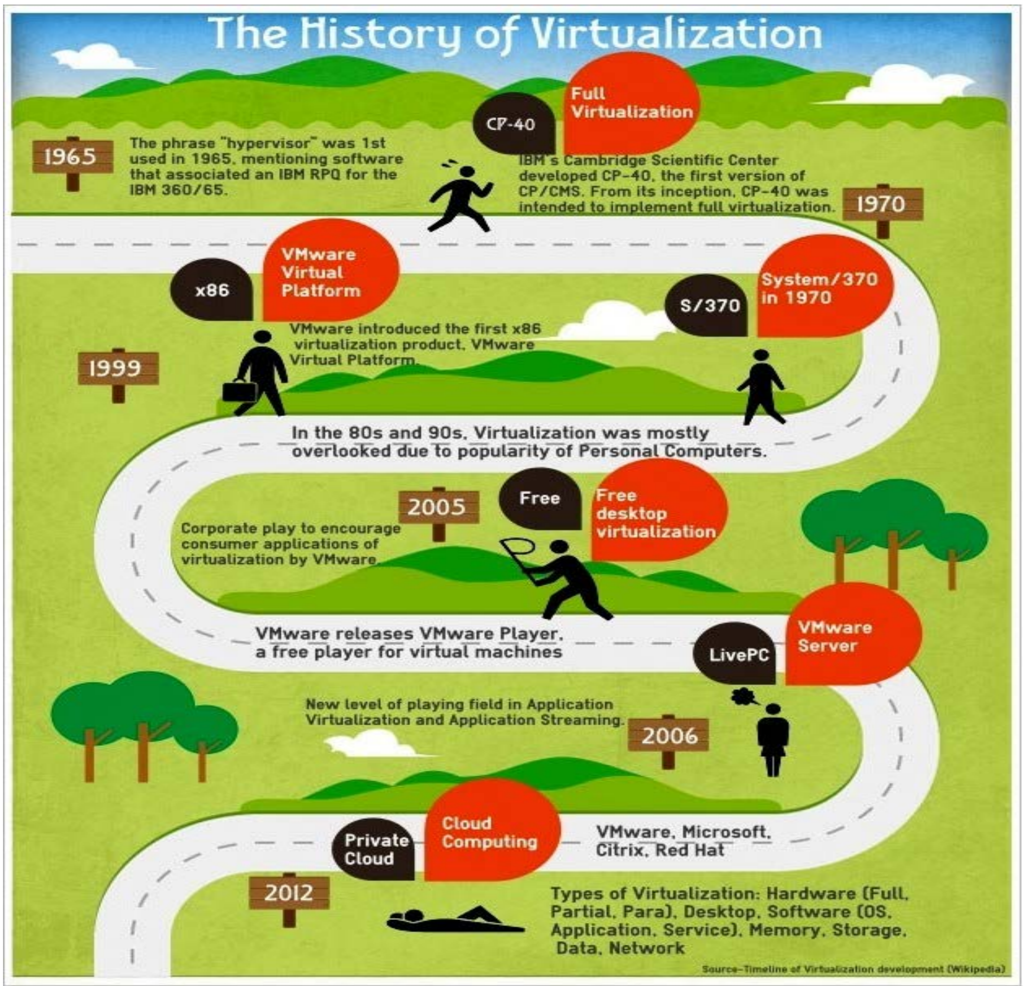
In 1972, an American computer scientist, Robert P. Goldberg introduced the theoretical basis of the architecture for virtual computer systems in his dissertation at Harvard University. In the same year IBM introduced a mainframe that was able to simultaneously run different operating systems under the supervision of a control program – hypervisor. The IBM System 370 was the first commercial computer entirely designed for virtualization, with which, the operating system CP / CMS allows you to run multiple instances simultaneously. This was followed by the IBM z / VM, which took advantage of hardware virtualization. The VM / CMS is highly regarded and widely distributed in the industry and academia. Several modern approaches to virtualization implementations are very unique for mainframe of computers from IBM.

Over the years, virtualization has begun to fall by the wayside due to creation of new client / server applications and the decline of the mainframe platform that lost power before the rise of the x86 platform. According to VMWare, the widespread adoption of Windows and Linux as the operating system on servers in the 1990s eventually established the x86 architecture as the industry standard.

Due to high costs for the purchase of a mainframe, companies began to acquire x86 servers according to demand, this is a process called low-end (several small machines doing the work of a large dedicated server). In this scenario, instead of having a high initial cost with the purchase of a mainframe, they opt for purchasing smaller servers according to need. The impact of this new strategy was to ensure a good deal of backlash against hardware scaling problems, and most of these servers were used for a single application. Thus, according to International Data Corporation, in each implementation of a typical x86 server, the roof of CPU usage was between 10 to 15% of the total capacity of this server.

The servers were oversized for the application that would perform, and as a result, ended up suffering the same problem of mainframes of the 1960s, that is, not all took advantage of their computing power, and were underutilized.

Then in 1999, VMware Inc. introduced the concept of virtualization on the x86 platform as a more efficient way to operate the equipment of the platform, taking advantage of x86 servers to provide a computational structure that would enable the full utilization of computational resources of these servers.



From 2005, processor manufacturers like Intel and AMD have given more attention to the need to improve hardware support in their products. Intel with its Intel VT and AMD with AMD-V. These hardware contains features which allow to exploit hypervisors that are used with the improved technique of virtualization (full virtualization) that make it easier to implement and enhance the performance.

4.0 Requirements for Virtualization

Meeting hardware requirements for virtualization remains something of an art form. It is important to give each virtual machine the hardware it needs, but it is also important not to waste resources by over provisioning virtual machines. This decreases the total number of virtual machines that can simultaneously run on a host server, which increases cost.

4.1 Virtual CPUs

For most processes, VMWare recommends a single virtual CPU for a virtual machine. This reduces management overhead on both the guest system and VMWare. A single virtual CPU however has the same impact as a single physical CPU. Only a single thread at a time can be executed. VMWare cannot parallelize something that is not executed in parallel by the VM. That means a VM with two virtual CPUs can never use more than two physical CPUs at the same time. That explains the dependencies on the power of physical hardware.

The number of virtual CPUs depends on the size of the setup. One should increase or decrease the cores based on their requirements.

4.2 Memory

VMWare will resource pool on available physical memory if not otherwise instructed. That means one can have four VMs with each having 4 GB memory and physical hardware with only 12 GB. Also, the hypervisor needs approximately 500 MB space of memory. The virtual host can instruct the virtual machine not to use a portion of the assigned memory. This is known as ballooning, and it lets VMWare dynamically assign available memory from one VM to other. This can cause the VM to swap out running processes or parts of them.

Swapping in general can cause harm to application performance. It is particularly harmful to java applications because memory is cycled through all the time, which is harmful. Thus the whole of a java program must always be in physical memory.

4.3 I/O and Network

The server writes a lot of data to disk for session recording, receives a lot of data, and also sends a lot of data to the database. Disk and network are still shared resources in a VM, so it is necessary to ensure there are enough resources reserved to the server.

There are some guidelines for implementing I/O and network on server.

i) Network switch needs to be separated for incoming traffic to the server.
ii) It is needed to make sure that all other traffic (database, client, disk) is taking a different physical network switch.

iii) Disks are mostly connected via SAN (Storage Area Network) to a virtual memory. It is needed to ensure the network switch can take the load and if possible, is dedicated to the usage of the SAN.

5.0 Type1 and Type 2 Hypervisors

Hypervisor technology helped drive innovation in the world of cloud computing. A hypervisor is a process that separates a computer's operating system and application from the underlying physical hardware. Usually done as software although embedded hypervisors can be created for things like mobile devices. The hypervisor drives the concept of virtualization by allowing the physical host machine to operate multiple virtual machines as guests to help maximize the effective use of computing resources such as memory, network bandwidth and CPU cycles.

Even though VMs can run on the same physical hardware, they are still logically separated from each other. This means that if one VM experiences an error, crash or a malware attack, it doesn't extend to other VMs on the same machine, or even other machines. VMs are also very mobile – because they are independent of the underlying hardware, they can be moved or migrated between local or remote virtualized servers a whole lot easier than traditional applications that are tied to physical hardware.

Hypervisors use a thin layer of code in software or firmware to allocate resources in real-time. One can think of the hypervisor as the traffic cop that controls I/O and memory management. There are two types of hypervisors: Type 1 and Type 2.

Type 1 hypervisors run directly on the system hardware. They are often referred to as a 'native' or 'bare metal' or 'embedded' hypervisors in vendor literature. Type 2 hypervisors run on a host operating system. When the virtualization first began to take off, type 2 hypervisors were most popular. Administrator could buy the software and install it on a server they already had. Type 1 hypervisor are gaining popularity because building the hypervisor into the firmware is proving to be more efficient. According to IBM, type 1 hypervisors provide higher performance, availability and security than type 2 hypervisors.

IBM recommends that type 2 hypervisors be used mainly on client systems where efficiency is less critical or on systems where support for a broad range of I/O devices is important and can be provided by the host operating system.

Type 1 hypervisor is often referred to as a hardware virtualization engine. It has better performance and greater flexibility because it operates as a thin layer designed to expose hardware resources to virtual machines(VMs), reducing the overhead required to run the hypervisor itself. Type 1 hypervisor is a function in and of itself. Servers that run type 1 hypervisors are often single-purpose servers that offer no other function. They become part of the resource pool and are designed specifically to support the operation of multiple applications within various VMs.

Examples of type 1 hypervisors are Xen, Oracle VM server for SPARC, Oracle VM server for x86, Microsoft Hyper-V. Examples of type 2 hypervisors are VMware Workstation, VMware Player, VirtualBox.

As mentioned earlier, a type 1 hypervisor is more efficient than a type 2 hypervisor, yet in many ways they both provide the same type of functionalities because they both run the same kind of VMs. One can usually move a VM from a host server running a type 1 hypervisor to one running a type 2 hypervisor and vice versa. A conversion maybe required but the process works.

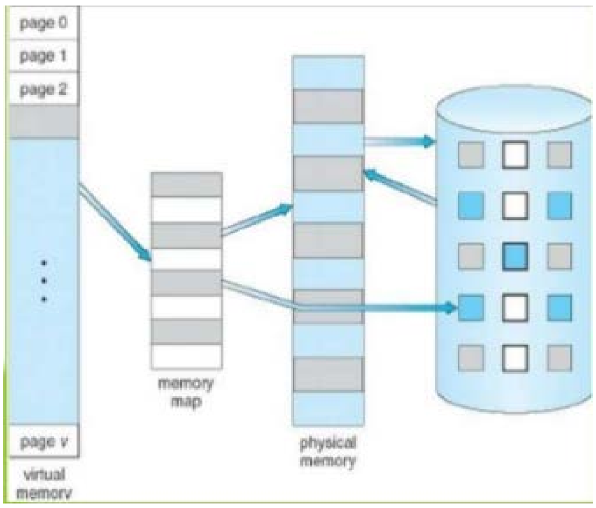
Because both types run directly on the hardware, type 1 hypervisors support hardware virtualization. Type 2 hypervisors perform software virtualization because they run as an application on top of an operating system.

6.0 Memory virtualization

Memory virtualization is seen as virtual memory, or swap on servers and workstations. It enhances performance by providing greater memory capacity, without the expense of adding main memory.

Memory Virtualization introduces a way to decouple memory from the processor and from the server to provide a shared, distributed or networked function. This is not more addressable memory, but virtualized memory shared between multiple machines.

When talking about Memory Virtualization, it is important to keep in mind that memory and storage are not synonymous. Memory Virtualization is focused on application performance and has a direct touch point with end users. The CPU actively and directly uses data from memory. On the other hand, storage is the persistent store of data. Storage is static, regardless of whether it is a spinning disk, SSD or RAM disk. Data is retrieved from disk and put into memory before the processor can transform data to information. Adding storage doesn’t solve memory problems and doesn’t noticeably accelerate application performance.



Now, where Memory Virtualization can be immediately applied across IT infrastructures.

- Extending memory beyond a physical server’s capacity.
- Implementing shared memory for clustered or grid computing environments.

- Enabling Cloud Computing and Real-Time Infrastructure (RTI) in the enterprise data center.

The choice to use a next generation data center or a Cloud Provider is an important strategic decision for any enterprise. Memory Virtualization plays an enabling role in both environments. Memory Virtualization is critical to implementing agile services for these infrastructures.

By offering memory as a pooled resource in the data center, Memory Virtualization delivers dramatic cost savings from reduced power and cooling. Real-time Infrastructure is available at 30% less cost in a pooled resource architecture versus over a tiered architecture.

Memory Virtualization is an enabler of Cloud Computing in this use case. Memory Virtualization enables different infrastructure Cloud Models to be available as different services with fungible resources. Memory resources underpin the MetaOS implementations (i.e. Microsoft Azure, Amazon EC2) that will govern these Clouds. Virtual memory provides the OS and application memory for this distributed operating system.

7.0 I/O virtualization

I/O virtualization (IOV), or input/output virtualization, is technology that uses software to abstract upper-layer protocols from physical connections or physical transports. This technique takes a single physical component and presents it to devices as multiple components. Because it separates logical from physical resources, IOV is considered an enabling data center technology that aggregates IT infrastructure as a shared pool, including computing, networking and storage.

In I/O virtualization, a virtual device is substituted for its physical equivalent, such as a network interface card (NIC) or host bus adapter (HBA). Aside from simplifying server configurations, this setup has cost implications by reducing the electric power drawn by these devices.

Virtualization and blade server technologies cram dense computing power into a small form factor. With the advent of virtualization, data centers started using commodity hardware to support functions such as burst computing, load balancing and multi-tenant networked storage.

I/O virtualization is based on a one-to-many approach. The path between a physical server and nearby peripherals is virtualized, allowing a single IT resource to be shared among virtual machines (VMs). The virtualized devices interoperate with commonly used applications, operating systems and hypervisors.

This technique can be applied to any server component, including disk-based RAID controllers, Ethernet NICs, Fibre Channel HBAs, graphics cards and internally mounted solidstate drives (SSDs). For example, a single physical NIC is presented as a series of multiple virtual NICs.

IOV enables enterprises to:

- Use existing cabling and peripheral components
- Improve server performance by using idle mezzanine slots
- Attach a single cable interconnect to support networking and storage I/O

- Reduce the cost of data center cooling, heating and power
- Scale for rapid redeployment as I/O profiles change

8.0 Virtual machines on Multicore CPUs

Multicore processing and virtualization are rapidly becoming ubiquitous in software development. They are widely used in the commercial world, especially in large data centers supporting cloud-based computing, to

(1) isolate application software from hardware and operating systems,

(2) decrease hardware costs by enabling different applications to share underutilized computers or processors,

(3) improve reliability and robustness by limiting fault and failure propagation and support failover and recovery, and

(4) enhance scalability and responsiveness through the use of actual and virtual concurrency in architectures, designs, and implementation languages.

Combinations of multicore processing and virtualization are also increasingly being used to build mission-critical, cyber-physical systems to achieve these benefits and leverage new technologies, both during initial development and technology refresh.

VMware multicore virtual CPU support lets you control the number of cores per virtual CPU in a virtual machine. This capability lets operating systems with socket restrictions use more of the host CPU's cores, which increases overall performance.

One can configure how the virtual CPUs are assigned in terms of sockets and cores. For example, you can configure a virtual machine with four virtual CPUs in the following ways:

i) Four sockets with one core per socket

ii) Two sockets with two cores per socket

iii) One socket with four cores per socket

Using multicore virtual CPUs can be useful when you run operating systems or applications that can take advantage of only a limited number of CPU sockets. Previously, each virtual CPU was, by default, assigned to a single-core socket, so that the virtual machine would have as many sockets as virtual CPUs.

With multicore SoCs, given enough processing capacity and virtualization, control plane applications and data plane applications can be run without one affecting the other. Data plane and control plane applications, in most cases, will be mapped to different cores in the multicore SoC as shown in Figure 2.

Control and data plane applications are not the only application level consolidation that will occur. Virtualization and partitioning will allow OEMs to enable their customers to customize service offerings by adding their own applications and operating systems to the base system on the same SoC, rather than using another discrete processor to handle it. Data or control traffic that is relevant to the customized application and operating system (OS) can be directed to the appropriate virtualized core without impacting or compromising the rest of the system.

Another example of consolidation of functions is board-level consolidation. Functions that were previously implemented on different boards now can be consolidated onto a single card and a single multicore SoC. Virtualization can present different virtual SoCs to the applications. With increasing SoC and application complexity, the probability of failures due to software bugs and SoC mis-configuration are greater than purely hardware-based failures. In such a paradigm, it may make sense to consolidate application-level fault tolerance onto a single multicore SoC, where a fraction of the cores are set aside in hot standby mode. While such a scheme will save the cost of having to develop a standby board or at the very least another SoC, it would require the SoC to be able to virtualize not only the core complex but also the inputs/ outputs (I/Os).

9.0 Recommendation

Virtualization and internet availability has increased virtualized server cluster or cloud computing environment deployments. With technological advances, faster network access with reduced latencies over internet is driving proliferation compute resource accesses or services using cloud computing model. This will further increase number of cloud or virtualized data centers and corresponding increase in energy consumption. Thus, energy efficient management of cloud or virtualized data center resources is crucial problem to reduce operating costs.

10.0 Conclusion

Proliferation of multicore processors in embedded markets and the desire to consolidate applications and functionality will push the embedded industry into embracing virtualization in much the same way as it occurred with the server and compute-centric markets.

Differences in the characteristics of embedded and compute-centric markets warrant different virtualization approaches. The embedded market, unlike the server and computecentric markets, is sensitive to the power envelope a particular device can dissipate and usually has very constrained power budgets as compared to those in the computecentric space. One of the primary design objectives in the embedded market is to maximize performance per watt, so it is desirable to offload as many functions to hardware as possible and free CPU cycles to be allocated to applications. When it comes to virtualization, the same philosophy is applied. While software-based solutions would work fine in the server and compute-centric markets, they are more likely to degrade the performance of an embedded system to unacceptable levels. In embedded markets, the bare-metal hypervisor-based approach coupled with hardware virtualization assists in the core, the memory subsystem and the I/O appears to offer the greatest performance over other approaches.

11.0 References

1) https://www.infoq.com/articles/virtualization-intro
2) http://pubs.sciepub.com/jcsa/2/3/1/
3) https://cloudtweaks.com/2012/12/the-history-of-virtualization/
4) https://www.vmware.com/my.html
5) https://community.dynatrace.com/community/display/DOCDT65/Virtualization+Requirements+and+Best+Practices