Volume 2, No. 12, December 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

VIRTUALIZATION TECHNIQUES & TECHNOLOGIES: STATE-OF-THE-ART

Rabi Prasad Padhy[*1], Manas Ranjan Patra[2], Suresh Chandra Satapathy[3]
*[1] Senior Software Engineer, Oracle Corporation, Bangalore, India
rabi.padhy@gmail.com
[2]Associate Professor, PG Department of Computer Science, Berhampur University, Berhampur, India
mrpatra12@gmail.com
[3]HOD & Professor, Department of Computer Science, ANITS, Sanivasala, India
sureshsatapathy@ieee.org

Abstract: Today virtualization is not just a possibility but it is becoming mainstream in data centers across the business world. Virtualization technology is popular today for hosting Internet and cloud-based computer services. Technology is evolving at a rapid rate and virtualization is no longer just about consolidation and cost savings, It is about the agility and flexibility needed for service delivery in data centers, including production environment and the infrastructure that supports the most mission-critical applications. Virtualization is a rapidly evolving technology that provides a range of benefits to computing systems, such as improved resource utilization and management, application isolation and portability and system reliability. Among these features, live migration, resources management, IT infrastructure consolidation are core functions. Virtualization technology also enables IT personnel to respond to business needs more rapidly with lower cost and improved operational efficiencies. Virtualization technologies are changing the IT delivery model to provide on-demand self-service access to a shared pool of computing resources via broad network.
The analysis of this survey will enable a better understanding of basic concept of virtualization, virtualization techniques, technologies, services and motivate especially Information Technologies (IT) staff as a detailed framework for the companies working in different industrial sectors. The evolution of datacenter transformation from traditional approach to virtualization i.e. from dedicated processing to pooled processing with strategic business values and methodologies are analyzed in details in this paper. This paper also discusses what virtualization is, how organizations can benefit from adopting virtualization into future IT plans.

Keywords: Virtual Machine, Virtual Machine Monitor, Virtualization, Server Consolidation, Virtual Infrastructure, Network Consolidation

INTRODUCTION

Virtualization can be considered IT asset optimization. It gaining popularity in enterprise environments as Infrastructure as a Service i.e. IaaS. For Data Centers, SMB and larger enterprises virtualization offers a solution for resource management (e.g., servers, storage devices, network devices, desktops and applications) which helps to achieve greater system utilization, lowering total cost and ease of management. Many experts agree that the current generation of enterprise information systems configured using dedicated resources and high ongoing support costs [1]. Virtualization also essential for implementation of Enterprise Grid and Cloud Computing. Virtualization layer hides the complexity and heterogeneity of the underlying hardware and software platforms and in effect creates a single virtual machine, operating a single virtual data source. Virtualization combines or divides the computing resources of a server based environment to provide different operating environments using different methodologies and techniques like hardware and software partition or aggregation, partial or complete machine simulation, emulation and time sharing. With the continuous evolution and improvements in virtualization technology, its adoption in various domains continues [2]. Virtualization is the technology that allows multiple operating system images running all at once by using only one piece of hardware.

Interesting architectures, platforms and applications have been designed, in order to take all benefits from such a novel approach in (parallel) computing – and the perspectives are obviously in a high spotlight. Enterprise today have over-provisioned hardware in their datacenters to accommodate surges in utilization and to ensure application isolation using dedicated servers. This has increased the total cost of ownership (TCO) of applications and reduce ROI on IT infrastructure. Virtualization is an efficient means of addressing these problems [3].

It is a technique through which hardware resources viz. processors, storage, I/O and networks on one or more machines can be transformed by hardware or software partitioning, time sharing and simulation / emulation of target machines into multiple execution environments each of which can act as a complete system by itself.

Virtualization allows multiple applications or operations to access and use the same resource while oblivious to accesses to the same resources by others. Virtualization provides access transparency and is a layer between the operating systems and the hardware. The operating systems or higher level applications are managed by a hypervisor or VMM. The VMM or hypervisor creates isolation paths as virtual machines where different operating systems run in virtual machines on top of the hypervisor. The hypervisor manages requests by virtual machines to access to the hardware.

DIFFERENT GENERATIONS OF VIRTUALIZATION

From 1950 to 1970:

The concept of virtual memory dates to the late 1950s when a group at the University of Manchester introduced automatic page replacement in the Atlas system, a transistorized mainframe computer. The principle of paging as a method to store and transmit data up and down the

memory hierarchy already existed but the Atlas was the first to automate the process, thereby providing the first working prototype of virtual memory [4].

The term virtual machine dates to the 1960s. One of the earliest virtual machine systems comes from IBM. Around 1967, IBM introduced the System/360 model 67, its first major system with virtual memory. Integral to the model 67 was the concept of a self -virtual zing processor instruction set, perfected in later models. The model 67 used a very early operating system called CP-67, which evolved into the virtual machine (VM) operating systems. VM allowed users to run several operating systems on a single processor machine. Essentially VM and the mainframe hardware cooperated so that multiple instances of any operating system, each with protected access to the full instruction set, could concurrently coexist.

In the mid 1960s IBM also pioneered the M44/44X project, exploring the emerging concept of time sharing. At the core of the system architecture was a set of virtual machines, one for each user. The main machine was an IBM 7044 (M44 for short) and each virtual machine was an experimental image of the 7044 (44X for short). This work eventually led to the widely –used VM/timesharing systems, including IBM's well-known VM/370. The concept of hardware virtualization also emerged during this time, allowing the virtual machine monitor to run virtual machines in an isolated and protected environment. Because the virtual machine monitor is transparent to the software running in the virtual machine, the software thinks that it has exclusive control of the hardware. The concept was perfected over time so that eventually virtual machine monitors could function with only small performance and resource overhead. By the mid 1970s, virtualization was well accepted by users of various operating systems.

The use of virtualization during these decades solved important problems. For example, the emergence of virtual storage in large-scale operating systems gave programs the illusion that they could address far more main storage (memory) than the machine actually contained. Virtual storage expanded system capacity and made programming less complex and much more productive. Also, unlike virtual resources, real system resources were extremely expensive. Virtual machines presented an efficient way to gain the maximum benefit from what was then a sizable investment in a company's data center. Although hardware - level virtual machines were popular in both the research and commercial marketplace during the 1960s and 1970s, they essentially disappeared during the 1980s and 1990s. The need for virtualization, in general, declined when low-cost minicomputers and personal computers came on the market.

From 1970 to 2000:

The period between 1970 and the early 1980s saw many new technologies and changes introduced to the computing world. IBM's mainframe operating systems were the primary drivers of enterprise computing innovation [5].

The IBM VM family introduced the concept of a hypervisor in 1972. Specifically, the core of the IBM VM family architecture was a control program (which effectively is what we now refer to as a hypervisor) called VM-CP. VM-CP ran on the physical hardware and created the virtual machine environment. This arrangement made VM-CP become classified as a type 1 hypervisor. Type 1 hypervisors are essentially software that runs directly on the hardware platform and thus below the operating system in the overall system stack. VM-CP provided complete virtualization of the physical machine, not just a subset as for previous solutions, and it really was the first implementation of both the hypervisor and the overall concept of virtualized machines.

The roots of virtualization on the mainframe begin in MVS (Multiple Virtual Storage was the most commonly used operating system on System/370 and System/390 machines. MVS was first introduced in 1974. The core of MVS has stayed the same throughout many revisions to this operating system), specifically with its virtual memory support. MVS allowed an unlimited number of applications to run in separate address spaces. The MVS core redirected requests from two separate applications for the same virtual memory address to different areas of physical memory. Additionally, for mainframes that had multiple processors configured in what was called ―loosely-coupled fashion, each processor had its own memory space and operating system to itself but shared peripherals connected to the physical box. A component of MVS called JES3 allowed this sharing and separation to be managed from a single console. While MVS is no longer supported by IBM, the roots of virtualization can be traced to its support of these two key capabilities. MVS transformed into OS/390 in 1995, which over time introduced support for 64-bit mainframe machines [6].

The IBM ESA/390 Operating System which Introduced Logical Partitions In 1990, IBM also introduced a logical partition, called an LPAR, for its ESA/390 mainframe architecture The LPAR is the core of mainframe virtualization. Another type of virtual machine, Sun Microsystems' Java Virtual Machine (JVM) and Microsoft's Common Language Runtime (CLR), deserve a place on the historical timeline and are worth mentioning here. The key thing to understand though is that these machines do not present a virtual hardware platform. These virtual machines emerged during the 1990s and extended the use of virtual machines into other areas, such as software development. Referred to as simulated or abstracted machines, they are implemented in software on top of a real hardware platform and operating system. Their beauty lies in their portability. In the case of JVM, compiled Java programs can run on compatible Java virtual machines regardless of the type of machine underneath the implementation.

From 2000 to 2011:

Virtualization has changed everything, for one machine can run a multitude of applications, each isolated into one virtual operating system, completely separated from the others. Finally, in 2000, OS/390 turned into z/OS, which is the current production IBM mainframe operating system. As a subsidiary of EMC, VMware popped onto the technology scene in 1998, beginning as a start-up company and soaring to become one of the driving forces in virtualization in just a few short years. In 2005, virtualization technology came in to mainstream awareness faster than anyone could have ever imagined - including many IT experts. Not only did it fly

right on through the developmental software stages, it fell headlong into the data centre.

VIRTUALIZATION CONCEPTS AND ARCHITECHTURES

What is Virtualization:

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

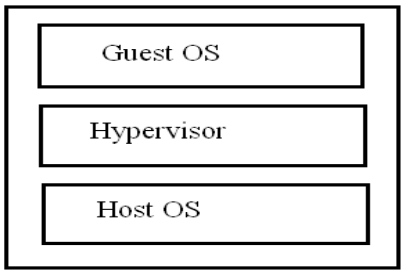

Figure 1: Basic Architecture of Virtualization

Virtualization is a system abstraction, in which a layer of virtualization logic manages and provides "virtualized" resources to a client layer running above it. The client accesses resources using standard interfaces, but the interfaces do not communicate with the resources directly; instead, the virtualization layer manages the real resources and possibly multiplexes them among more than one client. Virtualization is a combination of software and hardware engineering that creates Virtual Machines (VMs) - an abstraction of the computer hardware that allows a single machine to act as if it where many machines [7].

a. Without VMs: A single OS owns all hardware resources
b. With VMs: Multiple OS's, each running its own virtual machine, share hardware resources
c. Virtualization enables multiple operating systems to run on the same physical platform

Virtualization terminology:

Host Machine:

A host machine is the physical machine running the virtualization software. It contains the physical resources, such as memory, hard disk space, and CPU, and other resources, such as network access, that the virtual machines utilize.

Virtual Machine:

The virtual machine is the virtualized representation of a physical machine that is run and maintained by the virtualization software. Each virtual machine, implemented as a single file or a small collection of files in a single folder on the host system, behaves as if it is running on an individual, physical, non-virtualized PC.

Virtualization Software:

Virtualization software is a generic term denoting software that allows a user to run virtual machines on a host machine.

Components of virtualization:

Guest OS:

A guest OS is an operating system that runs in a virtual environment. A guest OS may be a client desktop, physical server or any other operating system that runs independently of dedicated hardware resources. Instead, the guest OS uses hardware resources allocated dynamically through a hypervisor or similar intermediary software.

Hypervisor or virtual machine manager (VMM):

A hypervisor also called a virtual machine manager (VMM), which is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. The task of this hypervisor is to handle resource and memory allocation for the virtual machines, ensuring they cannot disrupt each other, in addition to providing interfaces for higher level administration and monitoring tools [8].

A virtual machine monitor monitors a system of virtual machines (sometimes called hardware virtual machines), which allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. A virtual machine monitor monitors the software layer providing the virtualization, which is called a virtual machine. The VMM is the control system at the core of virtualization. It acts as the control and translation system between the VMs and the hardware.

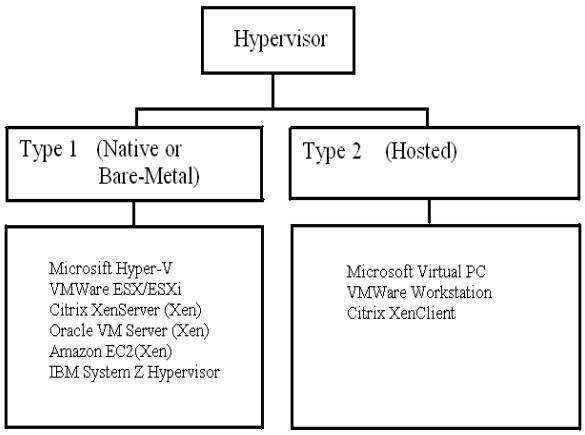

Figure 2: Hypervisor Types and Examples of products

A VMM is a layer of the most privileged code which has been abstracted from the rest of the OS. In the virtual machine architecture, this code is placed in the layer under the rest of the OS and specializes the functions of intercepting, monitoring and processing service calls from processes to the OS and the underlying hardware systems.

For example a cut-and-paste action from one process (e.g., a web page viewing application) to another (e.g. Document editor) will cause the following events in the OS: it first issues a service call for a read access to a memory location used by the former process, it then issues another call for a write access to a memory location of the latter process. The VMM is a thin software layer that runs directly on physical machines hardware. On top of the virtual machines monitor, there can be one or more virtual machines. The VMM provides each virtual machine with a set of virtual interfaces

that resemble direct interfaces to the underlying hardware. Applications on a virtual machine can run without modification as if they were on running on a dedicated physical machine [9]. The VMM allows multiple virtual machines to be running at the same time and transparently multiplexes resources between them. The VMM also isolated the virtual machines from one another, preventing them from accessing each other's memory or disk space. The operating system that runs inside of a virtual machine is traditionally referred to as the guest OS, and applications running on the guest OS are referred to as guest applications.

Virtual Machine Monitor (VMM) or Hypervisor can be categorized into two groups: Type I VMM and Type II VMM

Type I VMM:

a. It runs directly on the physical hardware
b. It does not have an OS running below it
c. Fully responsible for scheduling and allocating of the systems resources between virtual machines
d. Example : VMware ESX (Enterprise), Xen
e. More secure then Type II

Type II VMM:

a. VMM runs as an application in a normal OS.
b. The OS controls the real hardware resources called as Host OS.
c. Host OS has no knowledge of the type II VMM, which is treated like any other process in the system.
d. The OS runs inside of the Type II VMM is referred to as the Guest OS.
e. Example : VMware GSX (workstation), UML (User-Mode Linux)
f. Less secure then Type 1 because any security Vulnerabilities that lead to the compromise of the host OS will also give full control of the guest OS.
g. Host OS are heavyweight then Type II

Host OS:

A host OS (operating system) is the first OS installed on a machine to enable a machine to support multiple virtual operating systems. The host OS accesses the physical machine's resources, such as its physical memory and processor speed, and allocates those resources to virtual OS's as needed. A host OS is not necessarily the same as the OS that it hosts.

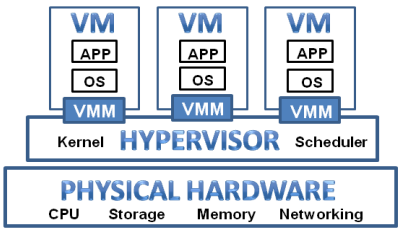

Figure 3: Basic components of Virtualized Infrastructure architecture

DIFFERENT METHODS OF VIRTUALIZATION

a. Full Virtualization
b. Para Virtualization
c. Emulation
d. Operating System Level Virtualization
e. Native Virtualization
f. Resource virtualization

Full Virtualization:

Full virtualization sometimes called hardware emulation. In this case an unmodified operating system is run using a hypervisor to trap and safely translate/execute privileged instructions on-the-fly. Because trapping the privileged instructions can lead to significant performance penalties, novel strategies are used to aggregate multiple instructions and translate them together. Other enhancements, such as binary translation, can further improve performance by reducing the need to translate these instructions in the future. Guest OS is unmodified and believes it is running on the same hardware as the host OS [10].

Para Virtualization:

Para virtualization also uses a hypervisor as like as full virtualization. It is the method where in a modified guest OS is able to speak directly to the hypervisor. This reduces translation time and overhead as the symbiotic relationship of the two is more efficient. However, unlike full virtualization, paravirtualization requires changes to the virtualized operating system. This allows the VM to coordinate with the hypervisor, reducing the use of the privileged instructions that are typically responsible for the major performance penalties in full virtualization. The advantage is that paravirtualized virtual machines typically outperform fully virtualized virtual machines. Para virtualization does not necessarily require the guest to look for the same hardware as the host, since the guest has limited communication with the host. However, the number of operating systems available for guest usage may be limited depending on the availability of paravirtualizationaware driver kits that must be used to ensure compatibility with the hypervisor.

Emulation:

It is a virtualization method in which a complete hardware architecture may be created in software. This software is able to replicate the functionality of a designated hardware processor and associated hardware systems. This method provides tremendous flexibility in that the guest OS may not have to be modified to run on what would otherwise be an incompatible architecture. Emulation features tremendous drawbacks in performance penalties as each instruction on the guest system must be translated to be understood by the host system. This translation process is extremely slow compared to the native speed of the host, and therefore emulation is really only suitable in cases where speed is not critical, or when no other virtualization technique will serve the purpose. Guest OS is unmodified but it is running on a software emulated CPU

Operating System Level Virtualization:

It is the method in which an operating system kernel provides for multiple isolated user-space instances. This is not true virtualization, however it does provide the ability for user-space applications (that would be able to run normally on the host OS) to run in isolation from other software. Most implementations of this method can define resource management for the isolated instances. The most

intrusive form of virtualization is operating system level virtualization. Unlike both paravirtualization and full virtualization, operating system-level virtualization does not rely on a hypervisor. Instead, the operating system is modified to securely isolate multiple instances of an operating system within a single host machine [11]. The guest operating system instances are often referred to as virtual private servers (VPS).

The advantage to operating system-level virtualization lies mainly in performance. No hypervisor/instruction trapping is necessary. This typically results in system performance of near-native speeds. The primary disadvantage is that all VPS instances share a single kernel. Thus, if the kernel crashes or is compromised, all VPS instances are compromised. However, the advantage to having a single kernel instance is that fewer resources are consumed due to the operating system overhead of multiple kernels. One of the good examples of OS level virtualization is Oracle Solaris containers.

Solaris container: Solaris container is a standard function of Solaris 10 and both sparc and x86 architectures are supported. It is an extension from "zone" on the previous version of Solaris. Solaris is acknowledged as robust and trust operating system. A new scheduler, Fair Share Scheduler (FSS), has been introduced to control running state of virtual operating system (zones). The FSS will prevent from locking of a CPU by a certain zone. The best advantage of Solaris container is that the patch management is easy. The enforcement of the patch to host operating system effects all zones. Moreover, zone can be pinned to specific CPU by standard function of Solaris. This function is effective under the symmetric multiple processor (SMP) environment.

Native Virtualization:

Native Virtualization techniques are another method for providing virtualized guests on a host system. This method does not seek to provide to the guests any hardware that is different from that of the host, thus any guest software must be host compatible. A software called a 'hypervisor' serves to translate the commands of the guest OS to the host hardware. A hypervisor may oversee several guest systems on a single host, and hypervisors are found in several virtualization methods. Native Virtualization lies as a middle ground between full emulation, and paravirtualization, and requires no modification of the guest OS to enhance virtualization capabilities. This compromise allows for an increase in speed (and indeed with hardware acceleration it can be very fast), but potential performance degradation can exist in an environment where the instructions are relying more heavily on the emulated actions rather than the direct hardware access portions of the hypervisor [12]. Native virtualization leverages hardware support for virtualization within a processor itself to aid in the virtualization effort. It allows multiple unmodified operating systems to run alongside one another, provided that all operating systems are capable of running on the host processor directly.

Native virtualization does not emulate a processor. This is unlike the full virtualization technique where it is possible to run an operating system on a fictional processor, though typically with poor performance. In x86 64 series processors, both Intel and AMD support virtualization through the Intel-VT and AMD-V virtualization extensions. x86 64 Processors with virtualization support are relatively recent, but are fast becoming widespread.

Resource virtualization:

Resource Virtualization is a method in which specific resources of a host system are used by the Guest OS. These may be software based resources such as domain names, certificates, etc. or hardware based such as shared storage space. This form of virtualization is largely used in the HPC (High Performance Computing) community because of it's advantages in forming a single logical computer across multiple nodes. Such a setup is beneficial when building cluster based supercomputers. In this way, the end user no longer needs to create cluster specific applications, instead the virtual machine can be treated like a single physical machine.

DIFFERENT TYPES OF VIRTUALIZATION

a. Server Virtualization
b. Storage Virtualization
c. Desktop / Client Virtualization
d. Network Virtualization
e. Application Virtualization

Server Virtualization:

Server virtualization is the most active segment of the virtualization industry featuring established companies such as VMware, Microsoft, and Citrix. A server is called as virtualized server when a single physical computing machine is made as multiple virtual machines. Each VM has its own virtual CPU, memory and peripheral interfaces and is capable of running its own OS by maintaining operational isolation and keeping security intact.

Each server typically serves one function (i.e., mail server, file server, Internet server, enterprise resource planning server, etc.), with each server using only a fraction of its true processing power, server virtualization breaks through the "one application, one server" barrier and facilitates the consolidation of numerous servers into one physical server. This equates to less physical servers required and 70 to 80 percent or higher utilization of existing hardware.

At the core of server virtualization is the concept of a hypervisor (virtual machine monitor). A hypervisor is a thin software layer that intercepts operating system calls to hardware. Hypervisors typically provide a virtualized CPU and memory for the guests running on top of them. The term was first used in conjunction with the IBM CP-370.

Hypervisors are classified as one of two types:

Type 1: This type of hypervisor is also known as native or bare-metal. They run directly on the hardware with guest operating systems running on top of them. Examples include VMware ESX, Citrix XenServer, and Microsoft's Hyper-V.
Type 2: This type of hypervisor runs on top of an existing operating system with guests running at a third level above hardware. Examples include VMware Workstation and SWSoft's Parallels Desktop.

Related to Type 1 hypervisors is the concept of paravirtualization. Paravirtualization is a technique in which software interface that is similar but not identical to the underlying hardware is presented. Operating systems must be ported to run on top of a paravirtualized hypervisor. Modified operating systems use the "hypercalls" supported by the paravirtualized hypervisor to interface directly with the hardware. The popular Xen project makes use of this type of virtualization. Starting with version 3.0 however Xen is also able to make use of the hardware assisted virtualization technologies of Intel (VT-x) and AMD (AMD-V). These extensions allow Xen to run unmodified operating systems such as Microsoft Windows [13].

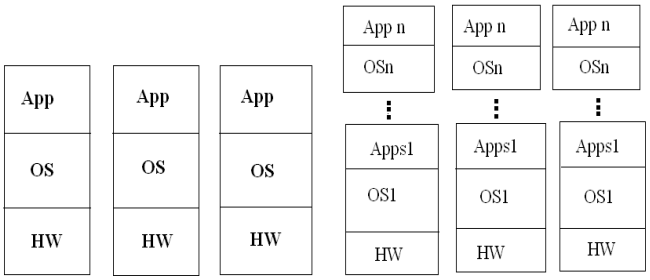

Figure 4: Basic difference Stand alone Servers and Virtualized Servers.

Server virtualization has a large number of benefits for the companies making use of the technology. Among those frequently listed:

Increased Hardware Utilization: This results in hardware saving, reduced administration overhead, and energy savings.

Security: Clean images can be used to restore compromised systems. Virtual machines can also provide sandboxing and isolation to limit attacks.

Development: Debugging and performance monitoring scenarios can be easily setup in a repeatable fashion. Developers also have easy access to operating systems they might not otherwise be able to install on their desktops.
Correspondingly there are a number of potential downsides that must be considered:

Security: There are now more entry points such as the hypervisor and virtual networking layer to monitor. A compromised image can also be propagated easily with virtualization technology.

Administration: While there are less physical machines to maintain there may be more machines in aggregate. Such maintenance may require new skills and familiarity with software that administrators otherwise would not need.
Licensing/Cost Accounting: Many software-licensing schemes do not take virtualization into account. For example running 4 copies of Windows on one box may require 4 separate licenses.

Performance: Virtualization effectively partitions resources such as RAM and CPU on a physical machine. This combined with hypervisor overhead does not result in an environment that focuses on maximizing performance.

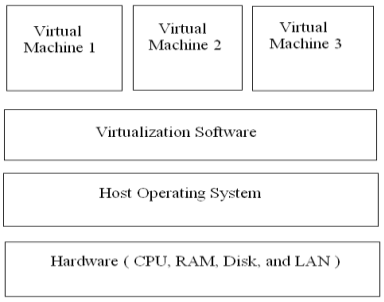

Figure 5: Basic Architecture of Server Virtualization

Amazon's EC2 Server (Elastic Compute Cloud 2) is a very good example of server virtualization.
Amazon's EC2: Amazon Elastic Compute Cloud is a web service that enables customers to lunch and manage Linux/Unix/Windows server instances in the data centers of Amazon [14].

Major features EC2:

a. Resource delivered as AMI (Amazon Machine Image)
b. Compute instance
c. Explicit access control

Advantages of Server Virtualization:

a. Reduce operational costs (Hardware, Energy, Space)
b. Improve uptime and availability
c. Enable robust Disaster Recovery
d. Reduce maintenance disruption
e. Streamline resource provisioning and scale
f. Server Virtualization enables server consolidation

Disadvantages of Server Virtualization:

a. Only a few processors that support virtualization can be used to virtualize servers.
b. The resource allocation for each virtual system needs to be planned carefully. If very less resource are allocated, the application performance might be affected and if too much resources are allocated, it will result in under-utilization.

Storage virtualization:

Storage virtualization is the pooling of multiple physical storage resources into what appears to be a single storage resource that is centrally managed. Storage virtualization is commonly used in file systems, storage area network (SANS), switches and virtual tape systems.

The act of abstracting, hiding, or isolating the internal function of a storage system, subsystem or service from applications, compute servers or general network resources for the purpose of enabling application and network independent management of storage or data. The application of virtualization to storage services or devices for the purpose of aggregating, hiding complexity or adding new capabilities to lower level storage resources. Storage can be virtualized simultaneously in multiple layers of a system, for instance to create Hierarchical storage management systems (HMS). Storage virtualization is a concept in System Administration, referring to the abstraction (separation) of logical storage (virtualized partitions of stored data) from physical storage (storage devices that hold, spin, read and write magnetic or optical disks such as CD, DVD, or even a

hard disk drive, etc.). This separation allows the Systems Admin increased flexibility in how they manage storage for end users [15].

Virtualization of storage helps achieve location independence by abstracting the physical location of the data. The virtualization system presents to the user a logical space for data storage and itself handles the process of mapping it to the actual physical location.

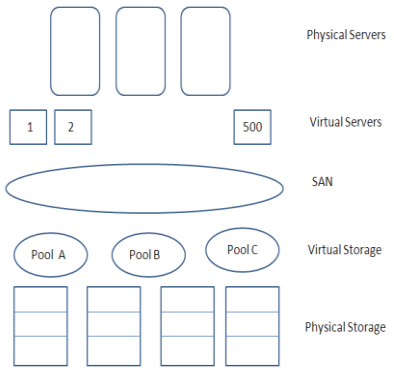

Figure 6: Basic Architecture of Storage Virtualization

There are three basic approaches to data storage:

Direct-Attached Storage (DAS): This is the traditional method used in data storage where hard drives are attached to a physical server. Because this method is easy to use but hard to manage, virtualization technology is causing organization to have a second thought with regard to its viability.

Network-Attached Storage (NAS): This is a machine that resides on your network and provides data storage to other machines. It can be thought of as the first step toward storage virtualization. This approach provides a single source of data, facilitating data backup. By collecting your data in one place, it also avoids the problem of multiple servers needing to access data located on another server.

Storage Area Network (SAN): This ultra-sophisticated approach deploys specialized hardware and software to transform mere disk drives into a data storage solution that transfers data on its own high-performance network. Companies shift over to a SAN when they recognize that corporate data is a key resource that must be available 24/7 and needs to be conveniently managed. The price tag for this approach is very high indeed.

The largest benefits of storage virtualization are:

a. Non-disruptive data migration
b. Centralized management
c. Increased utilization
d. Better visibility

Block virtualization: The act of applying virtualization to one or more block based (storage) services for the purpose of providing a new aggregated, higher level, richer, simpler, secure, etc., block service to clients. cf. file virtualization. Block virtualization functions can be nested. A disk drive, RAID system or volume manager all perform some form of block address to (different) block address mapping or aggregation.

Tape library virtualization: The act of creating abstracted tape devices by applying virtualization to tape drives or tape libraries.

File system virtualization: The act of aggregating multiple file systems into one large virtual file system. Users access data objects through the virtual file system; they are unaware of the underlying partitioning. The act of providing additional new or different functionality, e.g., a different file access protocol, on top of one or more existing file systems.
File virtualization: The use of virtualization to present several underlying file or directory objects as one single composite file. The use of virtualization to provide Hierarchical storage management systems (HSM) like properties in a storage system. The use of virtualization to present an integrated file interface when file data and metadata are managed separately in the storage system. cf. block virtualization [16].

Desktop virtualization:

Creates a separate OS environment over and above the existing running OS on the desktop. This allows a non compatible legacy or LOB application to operate within a more current desktop OS. There are two main variants of desktop virtualization:

Remote (Server-Hosted) Desktop Virtualization: In this model, the operating environment is hosted on a server in the data center and accessed by the end user across a network. Users connect to the server via connection brokers and receive their user interface via standard protocols, such as Remote Desktop Protocol. VMware Inc. is among the leaders in this market. Other vendors include Citrix Systems Inc., Virtual Iron Software Inc., and Qumranet Inc [17].

Local (Client-Hosted) Desktop Virtualization: In this model, the operating environment runs locally on the user's physical personal computer hardware and involves multiple flavors of client-side virtualization techniques that can monitor and protect the execution of the end user system. generally “hypervisor” software installed on the client device allows one desktop to run multiple operating systems. Top vendors in this market include VMware Inc., Microsoft Corp., Sentillion Inc., and Parallels Inc.

Virtualizing a desktop consists of removing the operating system from a traditional workstation and relocating a virtual copy of it on a host server. Users can then access the virtualized workstation, all of the programs, applications and data through a remote desktop client application from workstations, laptops, Smartphone’s or thin client terminals.
Desktop Virtualization Technologies:

a. Virtual Desktop Infrastructure (VDI): The server environment for desktop virtualization
b. Application Virtualization: software that runs programs virtually as though they are not on the local machine
c. Terminal Server/RD Session Host: MS remote desktop hardware application
d. Offline VDI: application that allows end-users to relocate their virtual machines to a local physical computer and back
e. Blade PC: PC that plugs into a server environment

Traditional Desktop:

a. Physical desktop or laptop
b. Locally installed OS
c. Locally installed applications

Virtual desktop infrastructure:

a. Physical or thin client computer
b. Hosted VM-based desktops
c. Applications local or virtualized
d. Connection using protocols like ica or rdp

Different Delivery models for desktop virtualization:

Server-side virtualization
a. Terminal services
b. Virtual hosted desktops

Client-side virtualization
a. Operating system image streaming
b. Application streaming and virtualization
c. Client-side virtual container

Hosted Shared Desktops
a. Session/presentation virtualization
b. Terminal Services/Citrix XenApp

Hosted VM Based Desktops (VDI)
a. Citrix XenDesktop
b. VMware View
c. MS VDI
d. Required Service Virtualization

Hosted Blade PC Desktops
a. Dedicated blades for each user

Locally Streamed Applications
a. Citrix XenApp
b. MS App-V(SoftGrid)
c. ThinApp

Locally Streamed Desktops
a. Ardence

Local VM Based Desktops
a. Type 1- Critrix XenClient
b. Type 2- VMware Player, VMware Workstation, MS Virtual PC.

Advantages of Desktop Virtualization:

Anywhere, Anytime Access: Empowers users with on-demand access to their virtual desktops when they need it, where they need it - from the office, home, road, customer site, etc [18].

Expedite Desktop Deployments: Enables swift, centralized desktop deployments, updates and maintenance—eliminating the hassle of local PC installations

Better Management and Control: Increases administrator's control over desktop configurations – while enabling desktop customization based on user needs

Enhanced Security: Desktops with applications and data are hosted within the datacenter – protecting sensitive information that would be compromised with stolen laptops or PCs

Full PC Desktop Experience: Virtual desktops maintain the same look and feel of traditional PCs – enabling a quick end user-migration to virtual desktops. Rich Integration with Existing Infrastructure support for various hypervisors including Xen, KVM and VMware.

Monetization and Back office Integration: Powerful back-office facing administrative API, enabling simple integration with providers' provisioning and billing systems and supporting automation of all administrative tasks.

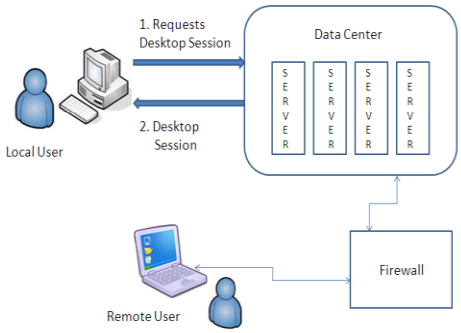

Figure 7: Basic Desktop Virtualization Architecture

Disadvantages of Desktop Virtualization:

a. Centralization means less customization for end users
b. Peripherals on older devices may be difficult to support
c. Multimedia applications can be impacted since virtualization solutions don't virtualize video card processors.
d. If the host server fails, users cannot access their virtual desktops.

Network virtualization:

Network virtualization proposes decoupling of functionalities in a networking environment by separating the role of the traditional Internet Service Providers (ISPs) into two: infrastructure providers (InPs), who manage the physical infrastructure, and service providers (SPs), who create virtual networks by aggregating resources from multiple infrastructure providers and offer end-to-end network services

A networking environment supports network virtualization if it allows coexistence of multiple virtual networks on the same physical substrate. Each virtual network (VN) in a network virtualization environment (NVE) is a collection of virtual nodes and virtual links. Essentially, a virtual network is a subset of the underlying physical network resources.

In a Network virtualization environment one or more virtual machines can access the local or external network using the physical network adaptor attached to physical machine. It can also be connected without physical network adaptor and it uses logical network adaptor. In Virtual machine if the physical adaptor is selected then it will get IP address from LAN, which allows it to communicate with network belongs to physical machine network. If it connects without any physical network adaptor then it will be part of internal virtual network [19].

Network virtualization is composed of two main components:

a. Link virtualization
b. Node virtualization

Link virtualization: Link virtualization allows the transport of multiple separate virtual links over a shared physical link. A virtual link is often identified explicit by a tag , but can

also be identified implicitly by a time slot or a wavelength, one of the good example is wide verity of the standard link virtualization techniques available in today's internet e.g. ATM, Ethernet 802.1q, MPLS) may in principle be used for this purpose.

Node virtualization: Node virtualization is based on isolation and partitioning of hardware resources. Physical resources of a substrate node (e.g. CPU, memory, storage capacity, link bandwidth) are partitioned into slice and each slice is allocated to a virtual node according to a set of requirements. Recent developments in OS virtualization have permitted significant advances in terms of fairness and performance.

Network Virtualization is the logical next step after storage and server virtualization. It allows multiple applications to run side-by-side over the same physical network. Each virtual network obeys business oriented policies while providing the security, availability and performance required for each service, from SAP and mail to VOIP and video. Virtual networks optimize the manageability and control of physical networks that are shared between multiple applications. The result is a quickly deployable, more reliable service that takes advantage of all the capabilities of the underlying hardware.

Network virtualization components:

a. Device virtualization: Virtualize physical devices in the network.
b. Data path virtualization: Virtualize communication path between network access points.
a) Hop-to-hop: Consider the virtualization applied on a single hop data path
b) Hop-top-cloud: Consider the virtualization tunnels allow multi-hop data path.
c) Protocols used to approach data-path virtualization.
   a. 802.1Q : Implement hop to hop data-path virtualization
   b. MPLS : Multiprotocol label Switch implement in router and switch layer virtualization
   c. GRE: Generic routing encapsulation implement in virtualization among variety of networks with tunneling technique.

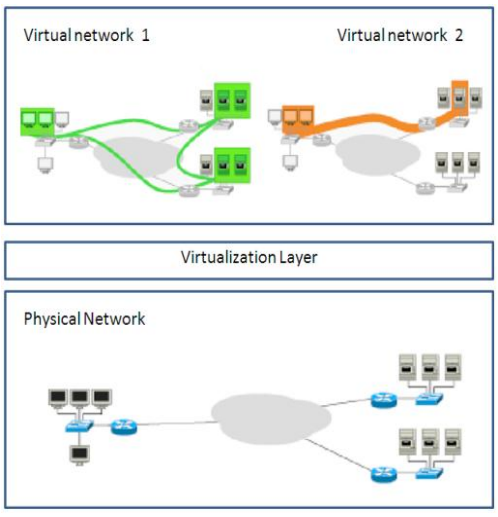

Figure 8: Basic Architecture of Network Virtualization

Types of network virtualization:

External network virtualization: In external network virtualization one or more local networks combined or subdivided into virtual networks. The goal is improving the efficiency of a large corporate network or data center. The main components of an external virtual network are the VLAN (Virtual Local Area Network) and the network switch. By using VLAN and switch technology, the system administrator can configure systems physically attached to the same local network into different virtual networks. VLAN technology also enables the system administrator to combine systems on separate local networks into a VLAN spanning the segments of a large corporate network.

Example: Cisco Systems' Service-Oriented Network Architecture which enables external network virtualization through use of the network switch hardware and VLAN software, or Hewlett Packard which has implemented external network virtualization through their X Blade Virtualization technologies.

Internal network virtualization: In Internal network virtualization, a single system is configured with containers, such as the Xen domain, combined with some hypervisor control programs or pseudo-interfaces such as the VNIC, to create a "network in a box." This solution improves overall efficiency of a single system by isolating applications to separate containers and pseudo interfaces.

Example: Open Solaris network virtualization which uses the "network in the box" scenario. It provides the ability for containers such as zones or virtual machines on a single system to share resources and exchange data.

The concept of multiple coexisting logical networks in the networking can be categorized into four main classes: VLANs, VPNs, active and programmable networks, and overlay networks.

a. Virtual Local Area Networks (VLAN)
b. Virtual Private Networks (VPN)
c. Active and programmable networks
d. Overlay networks

Virtual Local Area Networks (VLAN): A virtual local area network (VLAN) is a group of hosts with a common interest that are logically brought together under a single broadcast domain regardless of their physical connectivity. Since VLANs are logical entities, i.e., configured in software, they are flexible in terms of network administration, management, and reconfiguration. Moreover, VLANs provide elevated levels of trust, security and isolation, and they are cost-effective. Classical VLANs are essentially Layer 2 constructs, even though implementations in different layers do exist. All frames in a VLAN bear a common VLAN ID in their MAC headers, and VLAN-enabled switches use both the destination MAC address and the VLAN ID to forward frames. This process is known as frame coloring. Multiple VLANs on multiple switches can be connected together using trunking, which allows information from multiple VLANs to be carried over a single link between switches.

Virtual Private Networks (VPN): A virtual private network (VPN) is a dedicated communications network of one or more enterprises that are distributed over multiple sites and connected through tunnels over public communication networks (e.g., the Internet).Each VPN site contains one or more Customer Edge (CE) devices (e.g., hosts or routers), which are attached to one or more Provider Edge (PE) routers. Normally a VPN is managed and provisioned by a VPN service provider (SP) and known as Provider-provisioned VPN (PPVPN), while VPN implementations exist in several layers of the network stack, the following three are the most prominent ones.

a. Layer 3 VPN
b. Layer 2 VPN
c. Layer 1 VPN

Layer 3 VPN: Layer 3 VPNs (L3VPN) are distinguished by their use of layer 3 protocols (e.g., IP or MPLS) in the VPN backbone to carry data between the distributed CEs. L3VPNs can again be classified into two categories: CE-based and PE-based VPNs. In the CE-based VPN approach, CE devices create, manage, and tear up the tunnels without the knowledge of the SP network. In the PE-based approach, the provider network is responsible for VPN configuration and management. A connected CE device may behave as if it were connected to a private network.

Layer 2 VPN: Layer 2 VPNs (L2VPNs) provide end-to-end layer 2 connections between distributed cites by transporting Layer 2 frames between participating sites. The primary advantage of L2VPN is its support of heterogeneous higher-level protocols. But its lack of a control plane takes away its capability of managing reachability across the VPN.

Layer 1 VPN: The Layer 1 VPN (L1VPN) framework emerged from the need to extend L2/L3 packet-switching VPN concepts to advanced circuit-switching domains. It enables multiple virtual client-provisioned transport networks over a common Layer 1 core infrastructure. The fundamental difference between L1VPNs and L2 or L3 VPNs is that in L1VPNs data plane connectivity does not guarantee control plane connectivity and vice versa.

Active and programmable networks: Active and programmable networks may not be considered as direct instances of network virtualization, most of the projects in this area pushed forward the concept of coexisting networks through programmability. In order to allow multiple external parties to run possibly conflicting code on the same network elements, active and programmable networks also provide isolated environments to avoid conflicts and network instability. The programmable networks community discusses how communications hardware can be separated from control software. Two separate schools of thought emerged on how to actually implement such concepts: one from telecommunications community and the other from IP networks community.

Overlay networks: An overlay network is a virtual network that creates a virtual topology on top of the physical topology of another network. Nodes in an overlay network are connected through virtual links which correspond to paths in the underlying network. Overlays are typically implemented in the application layer, though various implementations at lower layers of the network stack do exist. Overlays are not geographically restricted, and they are flexible and adaptable to changes and easily deployable in comparison to any other network. As a result, overlay networks have long been used to deploy new features and fixes in the Internet.

Advantages of network virtualization: Minimize Downtime & Ensure Business Continuity: Virtualization isolates the network impact of multiple applications. Operating in the strict virtualization corset, applications do not interfere with each other and automated provisioning eliminates costly configuration errors.

Application virtualization:

This is a method of providing a specific application to an end user that is virtualized from the desktop OS and which is not installed in a traditional manner. An application can be installed and/or executed locally within a container that controls how it interacts with other system and application components. Or an application can be isolated in its own virtualized "sandbox" to prevent interaction with other system and application components. Or applications can be streamed across a network. Or applications can be delivered across the network to a web browser with most processing executed on a centralized web server. Application virtualization separates the application layer from the OS in a desktop environment which reduce application conflicts. By using this users can centrally manage patches and upgrades and accelerate the deployment of new applications. This also reduces the licensing costs [20].

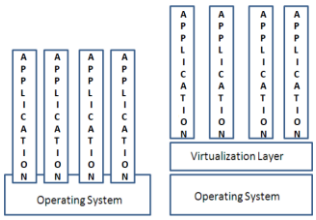

Figure 9: Basic Architecture of Application Virtualization

Advantages of Application Virtualization:

a. Reduce application-to-application conflicts
b. Reduce application compatibility regression testing
c. Centrally manage updates and patches
d. Faster software deployment
e. Full Portability: Virtualized applications can stream from any network share without a local client or a backend server.

Application virtualization products in market:

Company	Product
Xenocode	Virtual Application Studio
Microsoft	APP-V
VMWare	ThinApp

USAGE OF VIRTUALIZATION

Service Delivery: Automated self-service development and testing instead of slow, error-prone development and testing. A test bed can be created with virtual machines

Software evaluation: Untrusted software is evaluated in a virtual machine. The VM thus functions as a "sandbox" from which the software cannot escape

Running production applications: A business's applications are placed inside VM's

Desktop virtualization: rather then giving employees physical PC's, enterprises can provide them with a personal VM running on a central server.

Running cross-platform applications: an application developed for a specific OS is placed inside a VM, such that it can be run on a different OS (for example running a Windows application on a Mac).

Debug and replay: VMMs can replay and log actions of virtual machines. When a VM is infected with a virus, its actions can be studied, simply by replaying its execution.

Server Consolidation: To consolidation workloads of multiple under-utilized machines to fewer machines to save on hardware, management and administration of the infrastructure.

Application consolidation: A legacy application might require newer hardware and operating systems. Fulfillment of the need of such legacy applications could be served well by virtualizing the newer hardware and providing its access to others.

Sandboxing: Virtual machines are useful to provide secure, isolated environments (sendboxes) for running less-trusted applications virtualization technology can, thus help build secure computing platforms.

Multiple execution environments: Virtualization can be used to create multiple execution environments and can increase the QoS by guaranteeing specified amount of resources.

Virtual hardware : It can provide the hardware one never had, e.g. virtual SCSI drives, Virtual Ethernet adaptors, virtual Ethernet switches and hubs, and so on.
Software migration: Eases the migration of software and thus helps mobility.

Patch Management: Automated patching with no downtime instead of patch each host manually with downtime.

Live Migration: No maintenance window or planned downtime and migrate app in seconds instead of days and weeks with hardware maintenance window.

Appliances: Lets one package an application with the related operating environment as an appliance.

Instant Provisioning: Fully automated to days instead of weeks in a physical IT infrastructure environment.

Disaster Recovery: Automated testing during day and quick reliable restore instead weekend testing with uncertain restore.

SCOPE OF VIRTUALIZATION SERVICES

a. Server Consolidation
b. High Availability Disaster Recovery
c. Infrastructure Optimization
d. Infrastructure Automation
e. Client Virtualization
f. Software Lifecycle Management
g. Intelligent Infrastructure
h. Secure Computing
i. Applications

ADVANTAGES OF VIRTUALIZATION

a. Security: Generally there are different security requirements in different virtual machines. So one can select the guest OS and tools that are more appropriate for each environment. For example, we may want to run the Apache web server on top of a Linux guest OS and a backend MS SQL server on top of a guest Windows XP OS, all in the same physical platform. A security attack on one virtual machine does not compromise the others because of their isolation [21].
b. Reliability and availability: A software failure in a virtual machine does not affect other virtual machines.
c. Cost: It is possible to achieve cost reductions by consolidation smaller servers into more powerful servers. Cost reduction stem from into more powerful servers. Cost reductions stem from hardware cost reductions, operations cost reductions in terms of personnel, floor space and software licenses. VMware cites overall cost reductions ranging from 29 to 64%.
d. Adaptability to Workload Variations: Changes in work load intensity levels can be easily taken care of by shifting resources and priority allocations among virtual machines. Autonomic computing based resource allocation techniques can be used to dynamically move processors from one virtual machine to another.
e. Load Balancing: Since the software state of an entire virtual machine is completely encapsulated by the VMM, it is relatively easy to migrate virtual machines to other platforms in order to improve performance thought better load balancing.
f. Legacy Applications: Even if an organization decides to migrate to a different operating system, it is possible to continue to run legacy applications on the old OS running as a guest OS with in a VM. This reduces the migration cost.
g. Sustainability: Virtualized environments use less environmental resources. Energy consumption in data center is often wasted on machines that are consistently underutilized. Since virtualization allows for many virtual machines to run on one physical machine, less energy is needed to power and cool devices.
h. Responsiveness: Since the virtual environment has the ability to provision itself to get the best out of available

resources, response times are faster and downtimes can be reduced to near zero, improving agility and performance.

VARIOUS VIRTUALIZATION PRODUCTS & TECHNOLOGIES

a. VMware (GSX, ESX, VMware workstation)
b. Microsoft (Virtual PC, Virtual server, Hyper-v)
c. Open VZ (Open source container-based virtualization on Linux)
d. Sun (Solaris 10 containers)
e. HP (vPars, nPartions, IVM's)
f. IBM (PowerVM Virtualization)
g. VirtualBox (It's a open source)

Some of the Virtualization Software developed are listed below:

a. Wine
b. Disco
c. HP-UX Virtual Partitions
d. LPAR
e. Mac-on-Linux
f. Microsoft Virtual Server
g. Programming Language Virtual Machines include :
h. Solaris Zone Cluster
i. UML
j. VMware

Virtualization Solutions:

There are different virtualization solutions for different user types. It varies as per the requirement type as well as the user.

a. Virtualization solutions can help enterprise in consolidating servers and improve utilization
b. Provide on demand computing by scaling up/down the number of VM's/Applications
c. Reduce the number of servers required for fail over clustering , typically from 2*N to N+1 servers
d. Allow job/process migration smoothly across physical servers with near zero downtime.
e. Allow smaller enterprises to share servers across production and development environments without compromising the security of the production setup.
f. Continue to support legacy applications on older OS, while deploying the same along with newer application on newer OS versions on a single shared physical server.
g. Facilitate utility computing through deployments on computational grid.

VIRTUALIZATION & CLOUD COMPUTING

Virtualization forms the foundation of cloud computing, as it provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand. Cloud computing leverages virtualization technology to achieve the goal of providing computing resources as a utility. Cloud computing uses virtualization techniques to dynamically allocate and deallocate computing resources [22].

Virtualization is a key enabling technology for cloud computing environments. It is abstracting the IT resources. It is used by Cloud suppliers. Generally Cloud Computing is defined as a pool of virtualized computer resources. Based on this Virtualization the Cloud Computing paradigm allows workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines. A Cloud Computing platform supports redundant, self-recovering, highly scalable programming models that allow workloads to recover from many inevitable hardware/software failures. By means of virtualization technologies, Cloud computing offers to end users a variety of services covering the entire computing stack, from the hardware to the application level, by charging them on a pay per use basis. Some vendors, such as Amazon Web Services and VMWare base their offering on hardware level virtualization and provide bare compute and storage resources on demand. Google AppEngine and Microsoft Azure are more focused on application level virtualization by enforcing a specific application model that leverage their large infrastructure and scale up and down on demand.

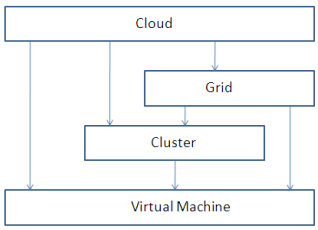

Figure 10: Overall Relational Model between Cloud, Grid, Cluster and Virtual Machine.

Amazon is one of the major players in providing IaaS solutions. Amazon Elastic Compute Cloud (EC2) provides a large computing infrastructure and a service based on hardware virtualization. By using Amazon Web Services, users can create Amazon Machine Images (AMIs) and save them as templates from which multiple instances can be run. It is possible to run either Windows or Linux virtual machines and the user is charged per hour for each of the instances running. Amazon also provides storage services with the Amazon Simple Storage Service (S3), users can use Amazon S3 to host large amount of data accessible from anywhere. With advances in virtualization technology, virtual machine services offered by cloud utility providers are becoming increasingly powerful, anchoring the ecosystem of cloud services [23].

In cloud models mainly on IaaS / HaaS customers are provided with virtualized hardware and storage on top of which they can build their infrastructure. Some of the examples are Amazon EC2 and S3, GoGrid, Nirvanix.

VIRTUALIZATION AND HIGH PERFORMANCE: BENEFITS AND CHALLENGERS

a. Fault-tolerance Migration: In HPC systems, the costs of application failure and restart are significant. With the move to petascale systems, these costs will rise further, requiring system developers to seek new reliability solutions (e.g., proactive faulttolerance. Virtualization is a key enabler for implementing the migration methods needed by fault-tolerance solutions, since each VM

cleanly encapsulates the entire application, library, and OS state to be migrated [24].

b. Fault-tolerance Monitoring: Pro-active fault-tolerance solutions require continuous system monitoring, including platform-level functionality for monitoring hardware components, application behavior and data integrity. Virtualization makes it possible to isolate application workloads and their needs from the control and management functionality needed to implement these solutions, without the need for specialized management hardware or software.

c. Shared I/O and service nodes: A key limiting factor for future machines is their ability to perform I/O. Here, virtualization can be of particular benefit to those nodes on high performance machines that are already (or that should be) shared by applications, as with I/O and service nodes. Their robustness can be improved by separating their onboard functionality for interacting with devices, for running system management tasks, and for doing so on behalf of different applications or application data streams (e.g., critical vs. non-critical I/O).

d. New functionality: Extended I/O services. Given the isolation mechanisms typically implied by virtualization, the I/O datapath can be extended with additional functionality. Examples include system level functionality like support for monitoring communication patterns, remote memory or device accesses, or system checkpointing,or higher, application-driven services, such as filtering, data staging, metadata management, etc. The former can be used to predict failures or capture undesirable performance behavior. Another example is to provide Quality of Service (QoS) support for separating the I/O streams of multiple or even single applications (e.g., preference given to critical checkpoints over additional I/O, desirable for faster restarts). The latter can provide improved services to applications, without requiring them to be rewritten or reorganized [25].

e. Portability and Manageability: Virtualization makes it possible to execute different application mixes along with required run-time systems in separate VMs on the same underlying physical platform. This enables end users to continue to use existing applications while upgrading to new systems or libraries or to easily compare new versions with older versions. Further, certain end users might simply continue to use older versions and systems, perhaps because of their reliance on ancillary functionality (e.g., for reliability management) that would otherwise be costly to port to newer system versions [26].

f. Development, debugging, and sharing: An interesting opportunity with virtualized systems is to 'debug at scale', that is, to create'test' partitions that extend across all nodes of the HPC machine, thereby permitting end users to scale their test runs to the sizes needed to validate scalability and/or other size-sensitive application behaviors. This can be done while at the same time making the machine available for other capacity runs. Another interesting opportunity is to strongly separate trusted from untrusted codes, again on the same underlying physical hardware.7. Mixed use for 'capability' and 'capacity' computing: The concurrency properties of future many-core platforms remain unclear, including their memory hierarchies, cache sharing across cores, redundancies and/or capacities of on-chip interconnects, core heterogeneity, etc. Virtualization provides a 'hedge' against the potentially negative (for large scale parallel applications) implications of such developments, making it possible to highly utilize a machine while applications are being improved, by enhancing existing machine partitioning methods with additional methods that partition the cores on individual nodes.

VIRTUALIZATION SECURITY CHALLENGES

Scaling: Virtualization enables rapid creation and addition of new virtual machines. Without total automation, this dynamic growth capacity can destabilize security management activities such as system configuration and updates, resulting in vulnerability to security incidents [27].

Transience: Whereas normal computing environments/networks tend to converge on a stable state, with a consistent collection of machines, virtualization environments can have machines that quickly come and go. This can foil attempts at consistent management, and leave, for instance, VMs that come and go and are vulnerable to and/or infected by a worm that goes undetected. Infections can persist within such a fluctuating environment and be difficult to stamp out.

a. Software lifecycle: Since a VM's state is encapsulated in the VMM software (along with any supporting hardware), snapshots of state can easily be taken. A VM can be instantiated from a prior snapshot, enabling easy state rollback this can interface with assumptions about the lifecycle of running software.

b. Diversity: Increased heterogeneity of operating systems and environments will increased security management difficulties, and present a more varied attack surface.

c. Identity: Static means of identifying machines, such as MAC addresses or owner name, may not function with virtualization. Machine ownership and responsibility is harder to track in a dynamic virtualized environment [28].

d. Data lifetime: Guest OS's may have security requirements about data lifetime that are invalided by a VMM's' logging with VM mobility, it is possible that sensitive data may be left in widely distributed persistent storage.

e. Integrity: Together with the feature of intervention and rollback comes the ability to manipulate the state of a VM, which threatens the integrity of the transactions done on a VM.

f. Non-Repudiation: Since virtual machines can be duplicated, rolled back and restored, there seems to be a fundamental problem regarding non-repudiation. If evidence of transactions is stored in a VM in the form

of a transaction log, this can be lost if the state in a VM is restored. If transactions are signed, the key with which this is done is also stored on the VM, and can thus be copied. Therefore, even if the digital signature appears valid, we are not entirely sure which machine actually put the signature.

CONCLUSION

In this paper we provided a detailed discussion of various virtualization techniques and technologies, the current state of affairs and a vision for the future. We also discussed a advantages of IT Infrastructure virtualization, multi-core systems and security. It is hoped that the article will provide researches many interesting avenues to explore in realizing the vision of highly scalable, well managed and green IT of the future.

REFERENCES

[1]. Heradon Douglas, Christian Gehrmann, Secure Virtualization and Multicore Platforms State-of-the-Art report, SICS Technical Report, Swedish Institute of Compute Sceince (December 2009).

[2]. Davide Adami, Stefano Giordano, Multidomain layer 1 Infrastructure Virtualization as a Feature Internet Services-enabling Paradigm, Journal of Internet engineering 4(1) (December 2010).

[3]. Sami Vaarala, Security Consideration of Commodity x86 Virtualization, PhD Thesis, Helsinki University of technology (May 2006).

[4]. Mueen Uddin, Azizah Abdul Rahman, Virtualization Implementation Model for Cost Effective & Efficient Data Centers, International Journal of Advanced Computer Science and Applications 2(1) (January 2011)69-74.

[5]. Erik van der kouwe, Anderw S. Tanenbaum, Vartual machine The State of the Art, Univesity of Amsterdam 30 July 2006.

[6]. Jenni Susan Reuben, A Survey on Virtual Machine Security, Helsinki University of Technology. Seminar on network Security October 2007.

[7]. Philip Reames, Ellick Chen, A Hypervisor for Embedded Computing, Illinois journal of undergraduate research (spring 2007).

[8]. Muhammad Bilal Anwer, Ankur Nayak, Network I/O Fairness in Virtual Machines, ACM journal (September 2010).

[9]. Haibo Chen, Jieyun Chen, Wenbo Mao, Fei Yan, Daonity Grid security from two levels of virtualization, Information security technical report 12(2007) 123-138.

[10]. Asma ben letaifa, Amed haji, Maha Jebalia, Sami Tabbane, State of the Art and Research Challenges of new services architecture technologies: Virtualization, SOA and Cloud Computing, International Journal of Grid and Distributed Computing 3(4) (December 2010) 69-88.

[11]. Tuncay Ercan, Towards virtualization: A competitive business continuity, African Journal of Business management 4(10) (August 2010) 2164-2173.

[12]. Daniel Gmach, Jerry Rolia, Resource pool management: Reactive versus proactive or lets be friends, Computer Networks 53(2009) 2905-2922.

[13]. Mueen Uddin, Azizah Abdul Rahman, Server Consolidation: An Approach to Make Data Center Energy Efficient & Green, International Journal of Scientific & Engineering Research 1(1) (October – 2010) 1-7.

[14]. Rabi Prasad Padhy, Manas Ranjan Patra, Suresh Chandra Satapathy, "RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's", International Journal of Advanced Engineering Science and Technologies, Vol. 11 (1) 15-30.

[15]. Yang Yu, OS-level Virtualization and its Applications, Stony Brook University, PhD thesis December 2007

[16]. Michail D. Flouris, Extensive networked-storage virtualization with Metadata management at the block level, PhD Thesis, University of Toronto (2009).

[17]. N.M. Mosharaf kabir Chowdhury, Raouf Boutaba, A survey of network virtualization, Computer Networks 54(2010) 862-876.

[18]. Fabio Baroncelli, Barbara Martini, Network virtualization for cloud computing, Ann. Telecommun journal (2010) 713-721.

[19]. N.M.M.K. Chowdhury, R.Boutaba, Network virtualization: state of the art and research challenges, IEEE Communications Magazine 47(7) (2009) 20-26.

[20]. Gurujit Singh Bhathal, G N Singh, A Comparative Study of Application Portability with Virtualization Software's. International Journal of Computer Science & Communication 1(2) (July-December 2010) 83-85.

[21]. Shtiaq Ali and Natarajan Meghanathan, Virtual Machines and Networks Installation, Performance, Study, Advantages and Virtualization Options, International Journal of Network Security & Applications 3(1) (January 2011) 1-15.

[22]. Flavio Lombardi, Roberto Di Pietro, Secure virtualization for cloud computing, Journal of Network and Computer Applications (2010) 1-10.

[23]. M.I.Jayalal, R.Jehadeesan, Moving From Grid to Cloud Computing: The Challengers in an Existing Computational Grid Setup, International Journal of Computer Science & Communication 1(2) (July-December 2010) 415-418.

[24]. S.Nagaprasad, A. VinayaBabu, K.Madhukar, Reviewing some platforms in cloud computing, International Journal of Engineering and Technology 2(5)(2010) 348-353.

[25]. Qi Zhang, Lu Cheng, Raouf Boutaba, Cloud Computing: State-of-the-art and research challenges, Journal of Internet server application (1) (2010) 7-18.

[26]. Mihai Christodorescu, Reiner Sailer, Cloud Security Is not (Just) Virtualization Security ACM journal (November 2009).

[27]. VMWare Documentation: http://www.vmware.com/ support/pubs/

[28]. Microsoft virtualization techcenter: http://technet.microsoft.com/

Short Bio Data for the Author

Rabi Prasad Padhy is currently working as a Senior Software Engineer - Oracle India Private Ltd. Bangalore. He has achieved his MCA degree from Berhampur University. He carries 8 years of extensive IT Experience with MNC's like EDS, Dell, IBM and Oracle. His area of interests include IT Infrastructure Optimization, Virtualization, Enterprise Grid Computing, Cloud Computing and Cloud databases. He has published several research papers in international journals. He is a certified professional for Oracle, Microsoft SQL Server database and also ITIL Certified.

Dr. Manas Ranjan Patra is heading the Post Graduate Department of Computer Science, Berhampur University in India. He holds a Ph.D. Degree in Computer Science from the Central University of Hyderabad, India. He has about 23 years of experience in teaching and research in different areas of Computer Science. He had visiting assignment to International Institute for Software Technology, Macao as a United Nations Fellow and for sometime worked as assistant professor in the Institute for Development and Research in Banking Technology, Hyderabad. He has about 75 international and national publications to his credit. His research interests include Service Oriented Computing, Software Engineering, Applications of Data mining and E-Governance. He has presented papers, chaired technical sessions and served in the technical committees of many International conferences.

Dr. Suresh Chandra Satapathy is a Professor and Head of the Dept of Computer Science and Engg. in Anil Nerrukonda Institute of Technology and Sciences (ANITS), Vishakapatnam, India. He has published many IEEE conference papers on Data clustering using PSO, GA etc. His areas of interest include Data mining, machine learning, Swarm Intelligence etc. He is a Senior Member of IEEE and Computer Society of India.