ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Analysis of Virtualization Tools and Techniques

Mrs. Kavita S.[1], Mrs. Dakshyani G.[2], Mrs. Kiruthika[3]

Assistant Professor, Department of Computer Engineering, FCRIT, Vashi, Navi Mumbai, India[1,2]

Associate Professor, Department of Computer Engineering, FCRIT, Vashi, Navi Mumbai, India[3]

Abstract: Virtualization is a framework of dividing the resources of a computer into multiple execution environments by applying one or more concepts or technologies such as hardware and software partitioning, timesharing, partial or complete machine simulation, emulation, quality of service, and many others. This technique allows user to share single physical instance of an application or resource among multiple users. This is done by assigning a logical name to a physical resource and providing address of that physical resource as when required. Discussion of cloud computing technique basically begins with virtualization. Virtualization is very important for computing as it simplifies the delivery of services by providing a platform for optimizing resources in a scalable manner, which makes cloud computing cost effective. This paper introduces tools and techniques used in virtualization and also its emphasis on virtual box tool.

Keywords: Virtual Machine(VM), Vitrual Box, Java Virtual Machine(JVAM)

I. INTRODUCTION

Virtualization benefits organizations by enabling users to share expensive hardware and software resources by multiplexing Virtual Machines (VMs). In this technology multiple VMs can run on a single hardware as if they are independent computers used by independent users. The idea of VMs originated in 1960. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

Hardware resources such as CPU, memory, I/O devices, and software resources like operating system and software libraries can be virtualized at various levels of abstraction.

A traditional computer runs with a host operating system especially for its hardware architecture, as shown in Fig 1. After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional Environment, called a Virtual Execution Environment in Fig 2.

Virtual Execution Environment is shown in the upper boxes, where applications run with their own guest OS over the virtualized CPU, memory, hard disk and I/O resource. Virtualization technologies find important applications over a wide range of areas such as server consolidation, secure computing platforms, supporting multiple operating systems, kernel debugging and development, system migration.

This technology handles platform diversity by running applications in VMs

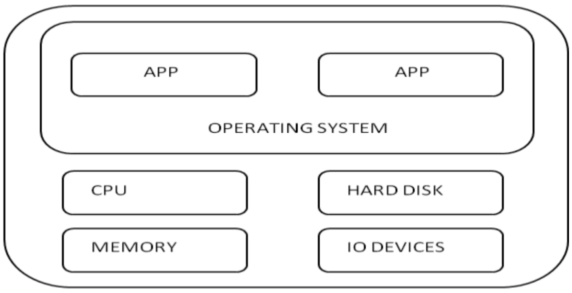

Fig 1: Real Computing Model

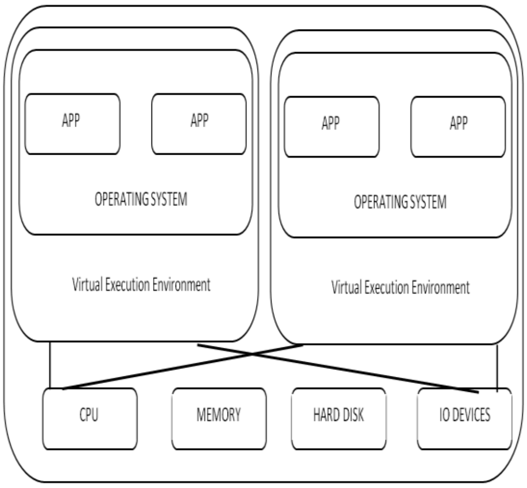

Fig 2: Virtual Computing Model

I. TYPES OF VIRTUALIZATION

A. *Infrastructure Virtualization*

i. *Network Virtualization*: It combines network hardware resources with software resources to provide virtualization of virtual network connection. It can be divided into VLAN and VPN.

ii. *Storage Virtualization*: It provides abstract logical view of physical storage device, so the user can access resources through logical interface of this view. It can be divided into storage device based storage Virtualization (e.g. RAID) and Network based Storage Virtualization (e.g. NAS, SAN).

DOI 10.17148/IJARCCE.2015.43134

B. System Virtualization

i) Core idea: One or more virtual machines using virtualization software on physical machine can be created.

ii) Desktop: Virtualization: Coupling relationship between PC desktop environment including applications and files, etc. and physical machines can be executed. Virtualized desktop environment is stored on a remote server, and when user has compatible device with sufficient display ability all the programs and data will eventually be stored in the remote server.

C. Software Virtualization

i) The High-level language virtualization: Migration problem of executable programs between different architectures can be solved. Programs which are written in high-level language will be compiled into standard intermediate instructions, and these instructions will be executed during interpretation or compilation.

ii) Application Virtualization : Applications can be decoupled from operating systems, and be provided a virtual running environment for applications, including application executable files and required runtime environment. Application virtualization server can push user required program components to the client virtual running environment timely (such as VMware ThinApp).

II. BENEFITS OF VIRTUALIZATION

The benefits of virtualization reach beyond the ability to take advantage of unused processor capacity. Here are some of those benefits: *f*

A. Resource utilization:
Virtualization provides economies of scale that allows existing platforms to be better utilized.

B. Infrastructure Cost Reduction:
It can reduce the number of servers and related hardware. *f*

D. Application and Network Design Flexibility:
Application programmers can design and build distributed applications that span systems and geography.

E. Operational Flexibility:
IT administrators can accomplish more with a single stroke in a virtual environment than the support and maintenance of multiple devices.

F. Portability:
Virtual Machines combined with the encapsulation of virtual data makes it easy to move virtual machines from one environment to another for maintenance, replication (disaster recovery) , resource utilization

G. System Availability: significant reductions in downtime for system maintenance can be achieved with virtualization. *f*

H. Recovery: In case of disaster, virtualized systems can be recovered more quickly.

III. APPLICATION OF VIRTUALIZATION

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

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

C. *Sandboxing:*
Virtual machines are useful to provide secure, isolated environments (sandboxes) for running foreign or less-trusted applications. Virtualization technology can, thus, help build secure computing platforms.

D. *Multiple execution environments*:
Virtualization can be used to create multiple execution environments (in all possible ways) and can increase the QoS by guaranteeing specified amount of resources.

E. *Virtual hardware*:
It can provide the hardware one never had, e.g. Virtual SCSI drives, Virtual Ethernet adapters, virtual Ethernet switches and hubs, and so on.

F. *Multiple simultaneous OS*:
It can provide the facility of having multiple simultaneous operating systems that can run many different kind of applications.

G. *Debugging*:
It can help debug complicated software such as an operating system or a device driver by letting the user execute them on an emulated PC with full software controls.

H. *Software Migration*:
It eases the migration of software and thus helps mobility.

I. *Appliances:*
It lets one package on an application with the related operating environment as an appliance.

J. *Testing/QA*:
It helps in producing arbitrary test scenarios that are hard to produce in reality and thus eases the testing of software.

IV. TOOLS OF VIRTUALIZATION

A. VMware

VMware[5] is a VM (virtual machine) platform which helps in execution of unmodified operating system on the host or user level application hardware . Operating system which is being executed with VMware may get crashed, reinstalled, rebooted or crashed without any on the application running on the host computer. VMware gives the separation of guest operating system from the real host operating system so that if the guest operating system fails then the physical hardware or the host machine does not suffer from the consequences

B. ThinApp

It is an agent less application virtualization solution that isolates applications from their underlying operating systems to eliminate application conflict and streamline application delivery and management.

C. JAVA VM

The concept of virtualization can be applied not only to subsystems such as disks but to an entire machine. To implement a virtual machine, developers add a software layer to a real machine to support the desired architecture. By doing so, a VM can circumvent real machine compatibility and hardware resource constraints.

JVM[1] is a virtual machine that runs Java byte code. This code is most often generated by Java compilers, although the JVM has also been targeted by compilers of other languages. Programs intended to run on a JVM must be compiled into a standardized portable binary format comprised of Java byte code (synonymous to instructions), which typically comes in the form of .class files. This binary is then executed by the JVM runtime which carries out emulation of the JVM instruction set by interpreting it or by applying a just-in-time Compiler (JIT). The JVM, in addition to providing the instruction set interpreter, also provides the operating environment (that OS typically provides in a native system) for the Java byte codes. Thus, the Java platform is a combination of a virtual machine along with an operating environment (JRE).

D. Wine

Wine[13] is an implementation of the Windows API, and can be used as a library to port Windows applications to Unix. The second meaning suggests that Wine does look like Windows to the Windows binaries (.EXE files) and emulates its behaviour very closely to that of Windows. In other words, Wine is a virtualization layer on top of X and Unix to export the Windows API/ABI, thus letting the Windows binaries run on top of it.

E. Jail

”Jail” [9] is a FreeBSD based virtualization software that provides the ability to partition an operating system environment, while maintaining the simplicity of UNIX ”root” model. The environment captured within a jail are typical system resources and data structures such as processes, file system, network resources, etc. In Jail, users with privilege find the scope of their requests to be limited to the Jail, allowing system administrators to delegate management capabilities to each virtual machine environment.

F. Ensim

Ensim’s [10][11]Virtual Private Server employs a similar virtualization technique for server consolidation, cost reduction, and to increase the efficiency in the way web sites are managed and sold. Ensim VPS ”virtualizes” a server’s native operating system so that it can be partitioned into isolated computing environments called virtual private servers. These virtual private servers operate independently of each other, just like a dedicated server. To the operating system, an Ensim VPS appears to be an application, and to applications `it’s indistinguishable from the native operating system. As a result, an Ensim VPS appears and operates as a physical server for the users

G. Bochs

Bochs[3] is an open-source x86 PC emulator written in C++ by a group of people lead by Kevil Lawton. It is a highly portable emulator that can be run on most popular platforms that include x86, PowerPC, Alpha, Sun, and MIPS. It can be compiled to emulate most of the versions of x86 machines including 386, 486, Pentium, Pentium Pro or AMD64 CPU, including optional MMX, SSE, SSE2, and 3DNow instructions. Bochs interprets every instruction from power-up to reboot, emulates the Intel x86 CPU, a custom BIOS, and has device models for all of the standard PC peripherals: keyboard, mouse, VGA card/monitor, disks, timer chips, network card, etc. Since Bochs simulates the whole PC environment, the software running in the simulation thinks as if it is running on a real machine (hence virtual machine) and in this way, supports execution of unmodified legacy software (e.g. Operating Systems) on its virtual machines without any difficulty

H. QEMU

QEMU [4] is a fast processor emulator that uses a portable dynamic translator. It supports two operating modes: user space only and full system emulation. In the earlier mode, QEMU can launch Linux processes compiled for one CPU on another CPU, or for cross-compilation and cross-debugging. In the later mode, it can emulate a full system that includes a processor and several peripheral devices. It supports emulation of a number of processor architectures that includes x86, ARM, PowerPC, and Sparc, unlike Bochs that is closed tied with the x86 architecture

I. Xen

Issues like QoS, security, and denial of service motivate the researchers in University of Cambridge come up with a modified architecture for virtualization, called Xen [7][8]. Xen exports a paravirtualized architecture in each of its VMs to maximize performance and resource isolation yet maintaining the same application binary interface (ABI) as commodity operating systems. Although it does require the operating systems to be ported, the porting effort is kept as low as possible. It aims at supporting around a hundred VM instances within a single physical machine within a reasonable performance hit.

J. Virtual PC

Microsoft’s Virtual PC [6], recently acquired from Connectix, is a product very similar to what is offered by VMware Workstation. It is based on the Virtual Machine Monitor (VMM) architecture and lets the user create and configure one or more virtual machines. Apart from the

DOI 10.17148/IJARCCE.2015.43134

features supported by Vmware, it provides two 559is tinguishing functionalities. It maintains an *undo disk* that lets the user easily undo some previous operations on the hard disks of a VM. This enables easy data recovery and might come handy in several circumstances. The other striking feature is *binary translation*, which it uses to provide x86 machines on Macintosh-based machines. The overall organization of the above tools are shown in Fig 3

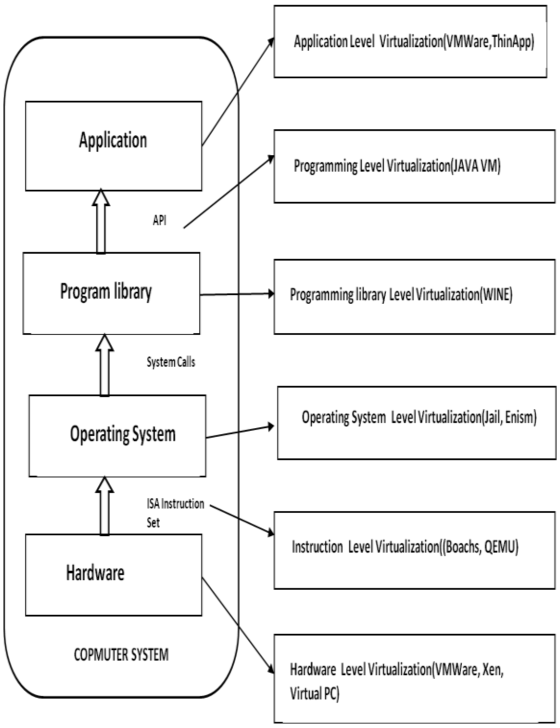
Application Level Virtualization(VMWare,ThinApp)
Application
Programming Level Virtualization(JAVA VM)
API
Program library
Programming library Level Virtualization(WINE)
System Calls
Operating System
Operating System Level Virtualization(Jail, Enism)
ISA Instruction Set
Instruction Level Virtualization((Boachs, QEMU)
Hardware
COPMUTER SYSTEM
Hardware Level Virtualization(VMWare, Xen, Virtual PC)

Fig. 3 Tools of Virtualization

V. VIRTUAL BOX

The techniques and features that Virtual Box provides are useful for several scenarios:

A. Running multiple operating systems simultaneously.
Virtual Box allows you to run more than one operating system at a time. This way, software written for one operating system on another (for example, Windows software on Linux or a Mac) can run without having to reboot to use it. Since one can configure what kinds of "virtual" hardware should be presented to each such operating system, one can install an old operating system such as DOS or OS/2 even if his/her real computer hardware is no longer supported by that operating system.

B. Easier software installations.
Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Virtual Box, such a complex setup can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Virtual Box.

C. Testing and disaster recovery. Once installed, a virtual machine and its virtual hard disks can be considered a "container" that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts. On top of that, with the use of another Virtual Box feature called "snapshots", one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong (e.g.after installing misbehaving software or infecting the guest with a virus), one can easily switch back to a previous snapshot and avoid the need of frequent backups and restores. Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.

D. Infrastructure consolidation. Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

E. Portability. Virtual Box runs on a large number of 32-bit and 64-bit host operating systems. No hardware virtualization required.

i). Guest Additions: shared folders, seamless windows, 3D virtualization. The Virtual Box Guest Additions are software packages which can be installed inside of supported guest systems to improve their performance and to provide additional integration and communication with the host system.

ii). *Great hardware support*: Among others, Virtual Box supports:

ii) *Guest multiprocessing (SMP):* Virtual Box can present up to 32 virtual CPUs to each virtual machine, irrespective of how many CPU cores are physically present on your host.

iv) *USB device support*: Virtual Box implements a virtual USB controller and allows you to connect arbitrary USB devices to your virtual machines without having to install device-specific drivers on the host. USB support is not limited to certain device categories.

v) Hardware compatibility: Virtual Box virtualizes a vast array of virtual devices, among them many devices that are typically provided by other virtualization platforms.
vi) Full ACPI support: The Advanced Configuration and Power Interface (ACPI) is fully supported by Virtual Box.

vii)Multiscreen resolutions. Virtual Box virtual machines support screen resolutions many times that of a physical screen, allowing them to be spread over a large number of screens attached to the host system.

DOI 10.17148/IJARCCE.2015.43134

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

VI. CONCLUSION

Virtualization tools is a wide topic of research since they are used in virtualization of hardware machines. In this paper, we discussed various types of virtualization, its benefits, various tools and techniques used in virtualization . Virtualization has a variety of applications and a few are discussed in this paper. Also the techniques and features of Virtual Box which are very useful in various scenarios are given emphasis.

REFERENCES

[1] Bill Venners, "The lean, mean, virtual machine," Java World, June 1996.
[2] R. Kessler and M. Hill, "Page placement algorithms for large real-indexed caches," in ACM Transaction on Computer Systems, November 1992, pp. 338–359.
[3] Kevin Lawton, Bryce Denney, N. David Guarneri, Volker Ruppert, Christophe Bothamy, and Michael Calabrese, "Bochs x86 pc emulator users manual," http://bochs.sourceforge.net/, 2003.
[4] "Qemu cpu emulator," http://fabrice.bellard.free.fr/qemu/qemu-tech.html, 2004.
[5] VMware, "Vmware workstation," http://www.vmware.com /products/deskto /ws features.html.
[6] Microsoft, "Microsoft virtual pc 2004," http://www.microsoft.com/windows/virtualpc/default.mspx.
[7] J. Navaro, S. Iyer, P. Druschel, and A. Cox, "Practical, transparent support for superpages," in ACM Operating Systems Review, Winter 2002 Special Issue, December 2002, pp. 89–104.
[8] R. Kessler and M. Hill, "Page placement algorithms for large real-indexed caches," in ACM Transaction on Computer Systems, November 1992, pp. 338–359.
[9] P. Kamp and R. Watson, "Jails: Confining the omnipotent root," in Proceedings of the 2nd International SANE Conference, 2000
[10] Ensim, "Virtual private servers," http://www.ensim.com/products/privateservers/index.html.
[11] Amit Singh, ," http://www.kernelthread.com.
[12] Bill Venners, "The lean, mean, virtual machine," Java World, June 1996.
[13] Wine Project, "Wine user guide," http://www.winehq.com /site/docs/wine-user/index.

DOI 10.17148/IJARCCE.2015.43134