Introduction

In 1998, VMware figured out how to virtualize the x86 platform, once thought to be impossible, and created the market for x86 virtualization. The solution was a combination of binary translation and direct execution on the processor that allowed multiple guest OSes to run in full isolation on the same computer with readily affordable virtualization overhead.

The savings that tens of thousands of companies have generated from the deployment of this technology is further driving the rapid adoption of virtualized computing from the desktop to the data center. As new vendors enter the space and attempt to differentiate their products, many are creating confusion with their marketing claims and terminology. For example, while hardware assist is a valuable technique that will mature and expand the envelope of workloads that can be virtualized, paravirtualization is not an entirely new technology that offers an "order of magnitude" greater performance.

While this is a complex and rapidly evolving space, the technologies employed can be readily explained to help companies understand their options and choose a path forward. This white paper attempts to clarify the various techniques used to virtualize x86 hardware, the strengths and weaknesses of each, and VMware's community approach to develop and employ the most effective of the emerging virtualization techniques. Figure 1 provides a summary timeline of x86 virtualization technologies from VMware's binary translation to the recent application of kernel paravirtualization and hardware-assisted virtualization.

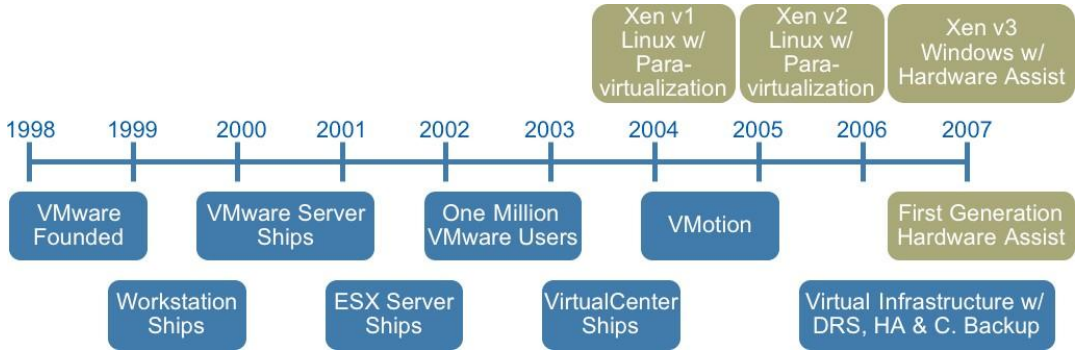

Figure 1 – Summary timeline of x86 virtualization technologies

Overview of x86 Virtualization

The term virtualization broadly describes the separation of a service request from the underlying physical delivery of that service. With x86 computer virtualization, a virtualization layer is added between the hardware and operating system as seen in Figure 2. This virtualization layer allows multiple operating system instances to run concurrently within virtual machines on a single computer, dynamically partitioning and sharing the available physical resources such as CPU, storage, memory and I/O devices.

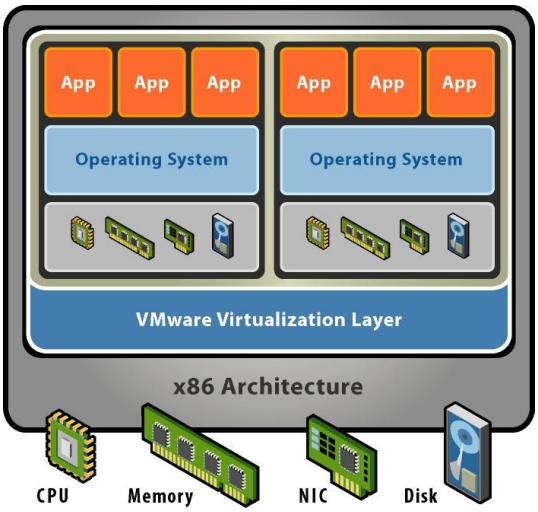

Figure 2 – x86 virtualization layer

As desktop and server processing capacity has consistently increased year after year, virtualization has proved to be a powerful technology to simplify software development and testing, to enable server consolidation, and to enhance data center agility and business continuity. As it turns out, fully abstracting the operating system and applications from the hardware and encapsulating them into portable virtual machines has enabled virtual infrastructure features simply not possible with hardware alone. For example, servers can now run in extremely fault tolerant configurations on virtual infrastructure 24x7x365 with no downtime needed for backups or hardware maintenance. VMware has customers with production servers that have been running without downtime for over three years.

For industry standard x86 systems, virtualization approaches use either a hosted or a hypervisor architecture. A hosted architecture installs and runs the virtualization layer as an application on top of an operating system and supports the broadest range of hardware configurations. In contrast, a hypervisor (bare-metal) architecture installs the virtualization layer directly on a clean x86-based system. Since it has direct access to the hardware resources rather than going through an operating system, a hypervisor is more efficient than a hosted architecture and delivers greater scalability, robustness and performance. VMware Player, ACE, Workstation and Server employ a hosted architecture for flexibility, while ESX Server employs a hypervisor architecture on certified hardware for data center class performance.

To better understand the techniques employed for x86 virtualization, a brief background on the component parts is useful. The virtualization layer is the software responsible for hosting and managing all virtual machines on virtual machine monitors (VMMs). As depicted in Figure 3Figure 3, the virtualization layer is a hypervisor running directly on

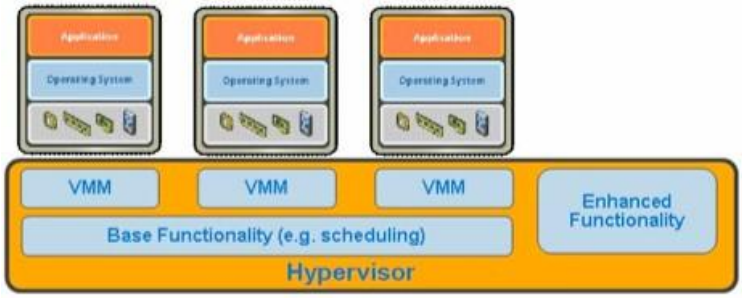

Figure 3 – The hypervisor manages virtual machine monitors that host virtual machines

the hardware. The functionality of the hypervisor varies greatly based on architecture and implementation. Each VMM running on the hypervisor implements the virtual machine hardware abstraction and is responsible for running a guest OS. Each VMM has to partition and share the CPU, memory and I/O devices to successfully virtualize the system.

CPU Virtualization

The Challenges of x86 Hardware Virtualization

X86 operating systems are designed to run directly on the bare-metal hardware, so they naturally assume they fully 'own' the computer hardware. As shown in Figure 4, the x86 architecture offers four levels of privilege known as Ring 0, 1, 2 and 3 to operating systems and applications to manage access to the computer hardware. While user level applications typically run in Ring 3, the operating system needs to have direct access to the memory and hardware and must execute its privileged instructions in Ring 0. Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0) to create and manage the virtual machines that deliver shared resources. Further complicating the situation, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0. The difficulty in trapping and translating these sensitive and privileged instruction requests at runtime was the challenge that originally made x86 architecture virtualization look impossible.

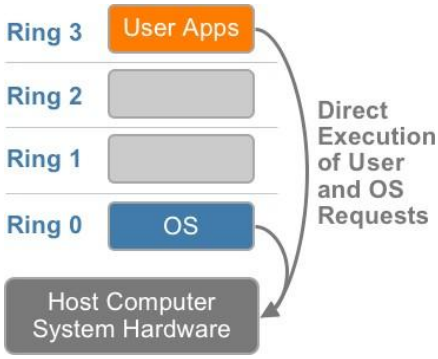

Figure 4 – x86 privilege level architecture without virtualization

VMware resolved the challenge in 1998, developing binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance, while moving the operating system to a user level ring with greater privilege than applications in Ring 3 but less privilege than the virtual machine monitor in Ring 0. While VMware's full virtualization approach using binary translation is the de facto standard today based on VMware's 20,000 customer installed base and large partner ecosystem, the industry as a whole has not yet agreed on open standards to define and manage virtualization. Each company developing virtualization solutions is free to interpret the technical challenges and develop solutions with varying strengths and weaknesses.

As clarified below, three alternative techniques now exist for handling sensitive and privileged instructions to virtualize the CPU on the x86 architecture:

- Full virtualization using binary translation
- OS assisted virtualization or paravirtualization
- Hardware assisted virtualization (first generation)

Technique 1 – Full Virtualization using Binary Translation

VMware can virtualize any x86 operating system using a combination of binary translation and direct execution techniques. This approach, depicted in Figure 5, translates kernel code to replace nonvirtualizable instructions with new sequences of instructions that have the intended effect on the virtual hardware. Meanwhile, user level code is directly executed on the processor for high performance virtualization. Each virtual machine monitor provides each Virtual Machine with all the services of the physical system, including a virtual BIOS, virtual devices and virtualized memory management.

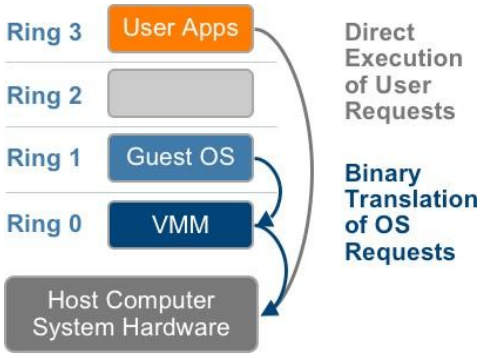

Figure 5 – The binary translation approach to x86 virtualization

This combination of binary translation and direct execution provides Full Virtualization as the guest OS is fully abstracted (completely decoupled) from the underlying hardware by the virtualization layer. The guest OS is not aware it is being virtualized and requires no modification. Full virtualization is the only option that requires no hardware assist or operating system assist to virtualize sensitive and privileged instructions. The hypervisor translates all operating system instructions on the fly and caches the results for future use, while user level instructions run unmodified at native speed.

Full virtualization offers the best isolation and security for virtual machines, and simplifies migration and portability as the same guest OS instance can run virtualized or on native hardware. VMware's virtualization products and Microsoft Virtual Server are examples of full virtualization.

Technique 2 – OS Assisted Virtualization or Paravirtualization

"Para-" is an English affix of Greek origin that means "beside," "with," or "alongside." Given the meaning "alongside virtualization," paravirtualization refers to communication between the guest OS and the hypervisor to improve performance and efficiency. Paravirtualization, as shown in Figure 6, involves modifying the OS kernel to replace non-virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor. The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling and time keeping.

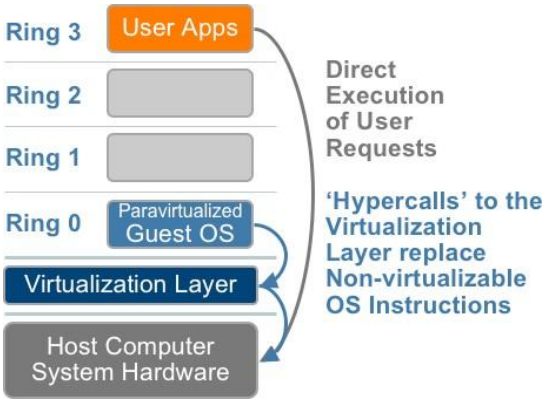

Figure 6 – The Paravirtualization approach to x86 Virtualization

Paravirtualization is different from full virtualization, where the unmodified OS does not know it is virtualized and sensitive OS calls are trapped using binary translation. The value proposition of paravirtualization is in lower virtualization overhead, but the performance advantage of paravirtualization over full virtualization can vary greatly depending on the workload. As paravirtualization cannot support unmodified operating systems (e.g. Windows 2000/XP), its compatibility and portability is poor. Paravirtualization can also introduce significant support and maintainability issues in production environments as it requires deep OS kernel modifications. The open source Xen project is an example of paravirtualization that virtualizes the processor and memory using a modified Linux kernel and virtualizes the I/O using custom guest OS device drivers.

While it is very difficult to build the more sophisticated binary translation support necessary for full virtualization, modifying the guest OS to enable paravirtualization is relatively easy. VMware has used certain aspects of paravirtualization techniques across the VMware product line for years in the form of VMware tools and optimized virtual device drivers. The VMware tools service provides a backdoor to the VMM Hypervisor used for services such as time synchronization, logging and guest shutdown. Vmxnet is a paravirtualized I/O device driver that shares data structures with the hypervisor. It can take advantage of host device capabilities to offer improved throughput and reduced CPU utilization. It is important to note for clarity that the VMware tools service and the vmxnet device driver are not CPU paravirtualization solutions. They are minimal, non-intrusive changes installed into the guest OS that do not require OS kernel modification. Looking forward, VMware is helping develop paravirtualized versions of Linux to support proofs of concept and product development. Further information is provided later in this paper on page 11.

Technique 3 – Hardware Assisted Virtualization

Hardware vendors are rapidly embracing virtualization and developing new features to simplify virtualization techniques. First generation enhancements include Intel Virtualization Technology (VT-x) and AMD's AMD-V which both target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0. As depicted in Figure 7, privileged and sensitive calls are set to automatically trap to the hypervisor, removing the need for either binary translation or

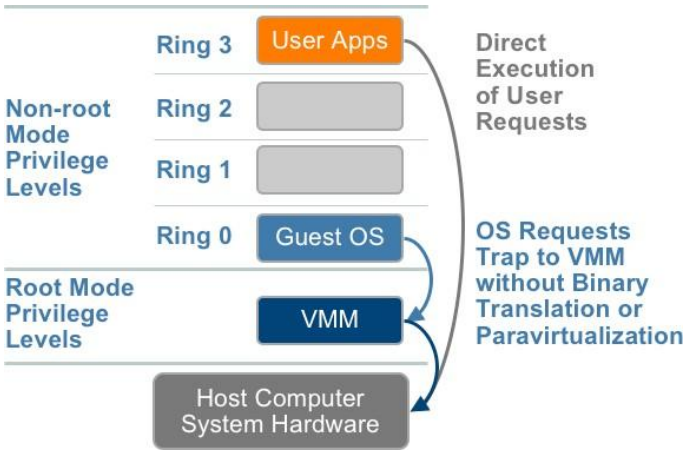

Figure 7 – The hardware assist approach to x86 virtualization

paravirtualization. The guest state is stored in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Blocks (AMD-V).

Processors with Intel VT and AMD-V became available in 2006, so only newer systems contain these hardware assist features.

Due to high hypervisor to guest transition overhead and a rigid programming model, VMware's binary translation approach currently outperforms first generation hardware assist implementations in most circumstances. The rigid programming model in the first generation implementation leaves little room for software flexibility in managing either the frequency or the cost of hypervisor to guest transitions[1]. Because of this, VMware only takes advantage of these first generation hardware features in limited cases such as for 64-bit guest support on Intel processors.

.

VMware is Fostering an Open Standards Approach to Virtualization

For the past several years, VMware has been collaborating with a group of leading technology vendors to define open virtualization standards. As an initial step, VMware has contributed its existing frameworks and APIs to facilitate the development of these standards in an industry neutral manner. VMware is proposing these open interfaces and formats because the most successful interfaces and formats in the technology business have been based on de facto customer deployed standards. VMware's technology has been deployed widely for over seven years and incorporates a significant amount of real-world experience.

In every industry, open interfaces and formats have proven to be a critical enabler to accelerating ubiquitous adoption, and virtualization is no different. As great as the momentum is today for virtualization, it is still in its early stages of adoption. VMware has taken this step to spur the growth of virtualization, accelerate solution delivery to customers and achieve widespread virtualization adoption.

VMware is also taking this step because partners and customers have asked for it. Open interfaces and formats benefit customers by providing access to a broader range of virtualization solutions that are compatible across an increased number of products. Open interfaces and formats benefit the industry by facilitating greater collaboration and innovation across an ecosystem of virtualization vendors to expand the market opportunities for all.

VMware has made available the following open interfaces and formats:

- **Virtual Machine Interface** — APIs between hypervisors and guest operating systems.
- **Management Interface** — Framework that governs the standardized operation and management of stand-alone virtual machine environments as well as highly dynamic, data center scale deployment of virtualized systems.
- **Virtual Machine Disk Format** — Virtual machine disk formats that enable virtual machine provisioning, migration and maintenance across platforms.

VMware intends this to be an open, vendor neutral effort. Any vendor that shares in the common goal of open virtualization standards can participate.

VMware Leverages a Multi-Mode VMM Architecture for Performance and Flexibility

Most startup virtualization vendors only have the resources to build a product that leverages a single strategy for virtualization. They naturally have a motivation to focus on the strengths of their virtualization approach and minimize their approach's weaknesses and feature gaps. This tends to generate market confusion as companies make one-sided claims that distort reality.

With each product release, VMware employs the combination of these and future virtualization technologies that strikes the most effective balance between performance, stability, functionality and ease of management. VMware also proactively works with partners to develop an interoperable ecosystem for enterprise computing virtualization.

VMware offers a flexible "multi-mode" VMM architecture depicted in Figure 12 that enables a separate VMM to host each virtual machine. VMware allows you to select the mode that achieves the best workload-specific performance based on the CPU support available. The same VMM architecture is used for ESX Server, Player, Server, Workstation and ACE.

While today's workloads can employ a 32-bit BT VMM or a 64-bit VMM with BT or VT-x, tomorrow's workloads will be hosted on VMMs that support 32 and 64-bit versions of AMD-V + NPT and VT-x + EPT.

VMware provides a flexible architecture to support emerging virtualization technologies. Multi-mode VMM utilizes binary translation, hardware assist and paravirtualization to select the best operating mode for each workload and processor combination. Hardware assist will continue to mature and broaden the workloads that can be readily virtualized.

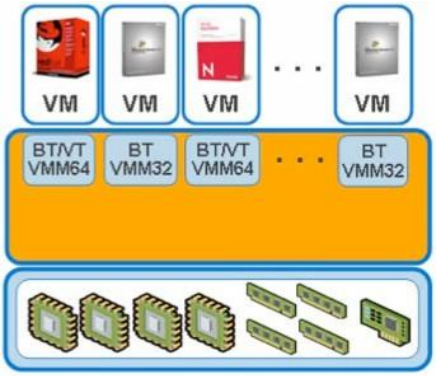

Figure 12 – Multi-mode VMM architecture

Conclusion

There is a range of strategies that VMware is pursuing to improve virtualization performance over time. Binary translation, hardware assist and OS assist (paravirtualization) are all effective techniques for x86 virtualization, but their relative value and importance will remain in flux as the enterprise computing virtualization market continues to evolve and mature. VMware started this revolution nearly a decade ago, and is continuing to lead the industry in building out an open standards and operating system agnostic virtualization ecosystem to help companies transform their IT environments.

Tomorrow's virtualization likely involves vendor-supported paravirtualized OSes that are installed into industry standard disk format files and able to run either natively or on a variety of compatible and interchangeable hypervisors that take advantage of hardware assisted management of the CPU, memory and I/O devices.

Today, VMware virtualization technology has been deployed by 100% of the Fortune 100 and 84% of the Fortune 1000 as the leading solution on the market. There is no alternative that compares to VMware's performance, stability, ease of management, security, support, features and vast partner ecosystem.

Next Steps

Enterprises evaluating computing virtualization should consider the value VMware Infrastructure can deliver. The VMware Sales Team can help your IT organization determine how VMware Infrastructure can benefit in your particular environment. Using consolidation assessments, ROI tools, case studies, and other tools, VMware will work with your team to design and implement specific success criteria so you can evaluate our software effectively. Visit us at www.vmware.com, email us at sales@vmware.com, or call us at 877-4VMWARE to get started.

Revision: 20070911 Item: WP-028-PRD-01-01

VMware, Inc. 3401 Hillview Ave. Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com

vmware®