Aalto University

Virtualization technology

Zhonghong Ou
Post-doc researcher
Data Communications Software (DCS) Lab,
Department of Computer Science and Engineering,
Aalto University

Definition

- Refers to a concept in which access to a single underlying piece of hardware, like a server, is coordinated so that multiple guest operating systems (OS) can share that single piece of hardware, with no guest OS being aware that it is actually sharing anything at all.
  - A guest OS is an OS that's hosted by the underlying virtualization software layer, which is often called the host system.

- A guest OS appears to the applications running on it as a complete OS, and the guest OS itself is completely unaware that it's running atop a layer of virtualization software rather than directly on the physical hardware.

Source: Virtualization for dummies

Four drivers of virtualization

- Hardware is underutilized
  - Moore's Law: "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year…", (most people estimate the timeframe at around 18 months) i.e. the number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years. Moore's law

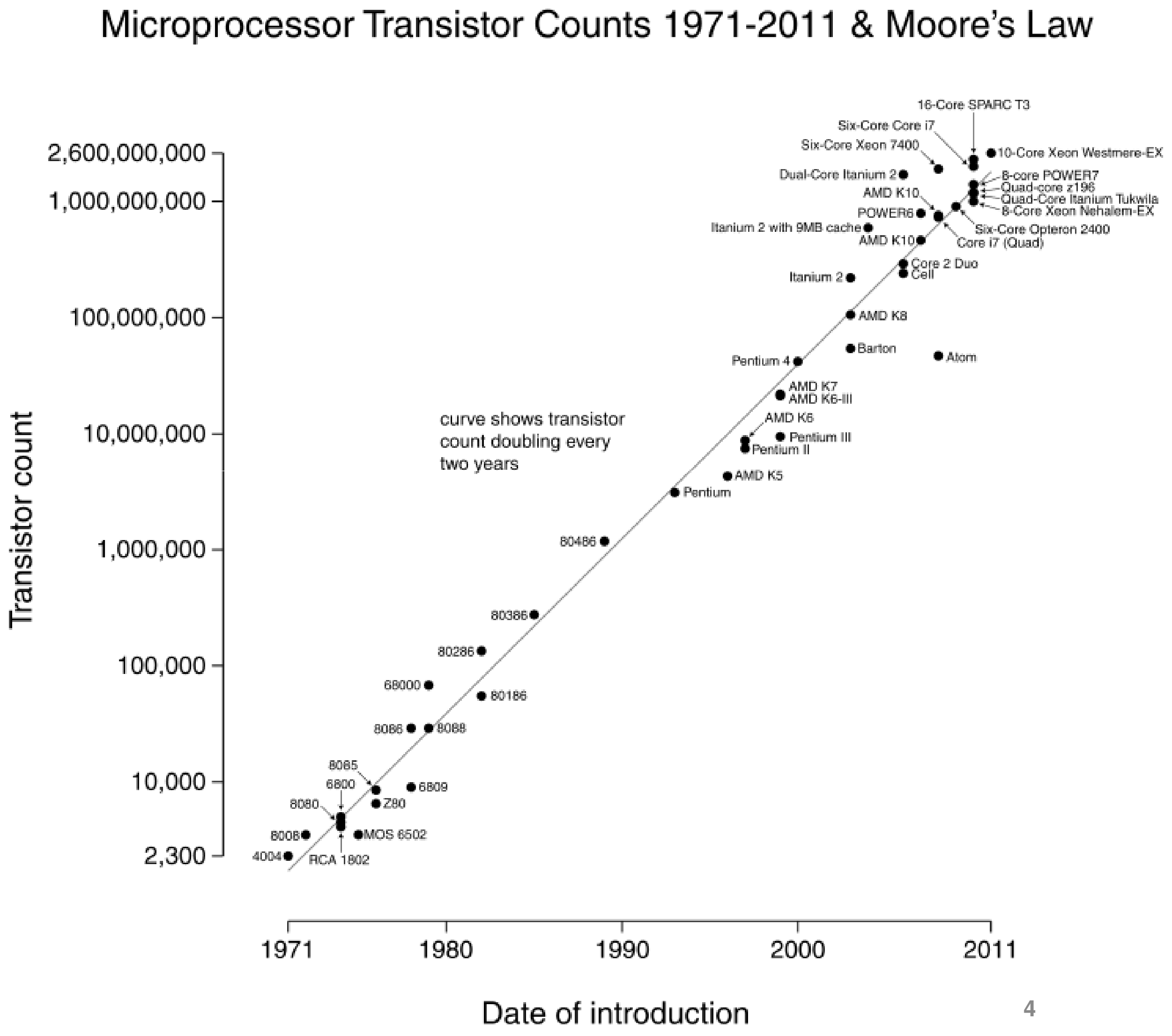
- Data centers run out of space
  - The net effect of explosion of data is that huge numbers of servers have been put into use over the past decade, which is causing a real-estate problem for companies.

- Energy costs go through the roof
  - The cost of running computers, coupled with the fact that many of the machines filling up data centers are running at low utilization rates, means that virtualization's ability to reduce the total number of physical servers can significantly reduce the overall cost of energy for companies.

- System administration costs mount
  - Computers don't operate on their own. Every server requires care and feeding by system administrators who, as part of the operations group, ensure that the server runs properly.

Source: Virtualization for dummies

Moore's law


Microprocessor Transistor Counts 1971-2011 & Moore's Law
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
Date of introduction
1971
1980
1990
2000
2011
curve shows transistor count doubling every two years
16-Core SPARC T3
Six-Core Core i7
Six-Core Xeon 7400
10-Core Xeon Westmere-EX
Dual-Core Itanium 2
8-core POWER7
Quad-core z196
Quad-Core Itanium Tukwila
8-Core Xeon Nehalem-EX
AMD K10
POWER6
Itanium 2 with 9MB cache
Six-Core Opteron 2400
Core i7 (Quad)
AMD K10
Core 2 Duo
Cell
Itanium 2
AMD K8
Barton
Atom
Pentium 4
AMD K7
AMD K6-III
AMD K6
Pentium III
Pentium II
AMD K5
Pentium
80486
80386
80286
68000
80186
8086
8088
8085
6800
6809
8080
Z80
8008
MOS 6502
4004
RCA 1802

Moore’s law

- Moore's law describes a long-term trend in the history of computing hardware. The number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years. The trend has continued for more than half a century and is not expected to stop until 2015 or later. The law is named after Intel co-founder Gordon E. Moore, who described the trend in his 1965 paper.

- Other formulations and similar laws
  - Transistors per integrated circuit. The most popular formulation is of the doubling of the number of transistors on integrated circuits every two years.
  - Density at minimum cost per transistor. This is the formulation given in Moore's 1965 paper. It is not just about the density of transistors that can be achieved, but about the density of transistors at which the cost per transistor is the lowest.
  - Power consumption. The power consumption of computer nodes doubles every 18 months.
  - Hard disk storage cost per unit of information. A similar law (sometimes called Kryder's Law) has held for hard disk storage cost per unit of information. The current rate of increase in hard drive capacity is roughly similar to the rate of increase in transistor count. Recent trends show that this rate has been maintained into 2007.
  - Network capacity. According to Gerry/Gerald Butters, the former head of Lucent's Optical Networking Group at Bell Labs, there is another version, called Butter's Law of Photonics, a formulation which deliberately parallels Moore's law. Butter's law says that the amount of data coming out of an optical fiber is doubling every nine months.
  - Pixels per dollar. Similarly, Barry Hendy of Kodak Australia has plotted the "pixels per dollar" as a basic measure of value for a digital camera, demonstrating the historical linearity (on a log scale) of this market and the opportunity to predict the future trend of digital camera price, LCD and LED screens and resolution.

Source: https://secure.wikimedia.org/wikipedia/en/wiki/Moore%27s_law

Moore’s law (cont.d)

- The Great Moore's Law Compensator (TGMLC)
  - Generally referred to as bloat, is the principle that successive generations of computer software acquire enough bloat to offset the performance gains predicted by Moore's Law.
  - In a 2008 article in InfoWorld, Randall C. Kennedy, formerly of Intel, introduces this term using successive versions of Microsoft Office between the year 2000 and 2007 as his premise. Despite the gains in computational performance during this time period according to Moore's law, Office 2007 performed the same task at half the speed on a prototypical year 2007 computer as compared to Office 2000 on a year 2000 computer.

Consolidation ratio

- Today 4:1(four virtual machines on one physical server) to 8:1;
- As organizations become more familiar with the technology and develop operational expertise, this ratio is expected to quickly grow to from 10:1 to 20:1.
- With the advent of multi-core CPUs, virtual machine densities are expected to approach 50:1 within 2 years.

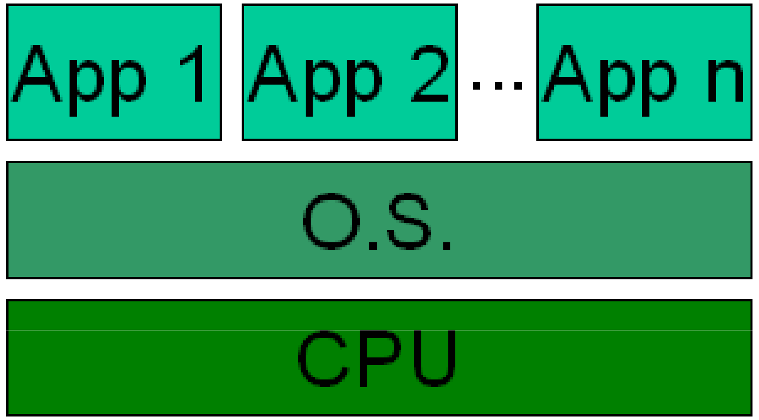
Source: Cisco

Reasons for moving to virtualization

- Saves money: Virtualization reduces the number of servers you have to run, which means savings on hardware costs and also on the total amount of energy needed to run hardware and provide cooling.
- Good for the environment: Virtualization is a green technology through and through. Energy savings brought on by widespread adoption of virtualization technologies would negate the need to build so many power plants and would thus conserve our earth's energy resources.
- Reduces system administration work: With virtualization in place, system administrators would not have to support so many machines and could then move from firefighting to more strategic administration tasks.
- Gets better use from hardware: Virtualization enables higher utilization rates of hardware because each server supports enough virtual machines to increase its utilization from the typical 15% to as much as 80%.
- Makes software installation easier: With software vendors tending more and more towards delivering their products preinstalled in virtual machines (also known as virtual appliances), much of the traditional installation and configuration work associated with software will disappear.

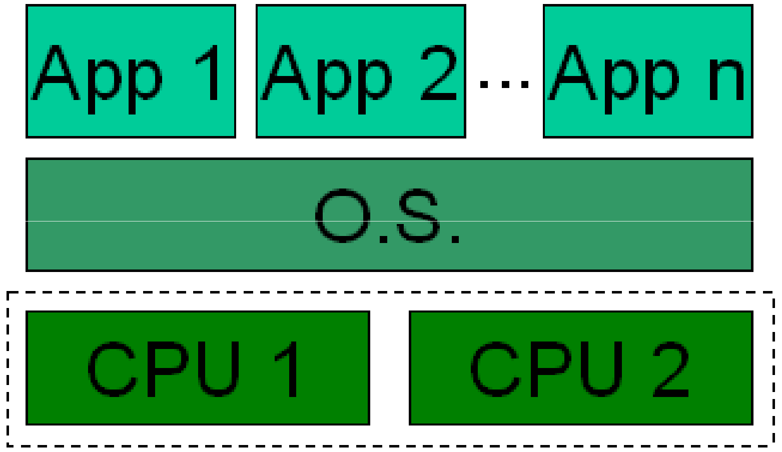
Source: Virtualization for dummies

Multitasking



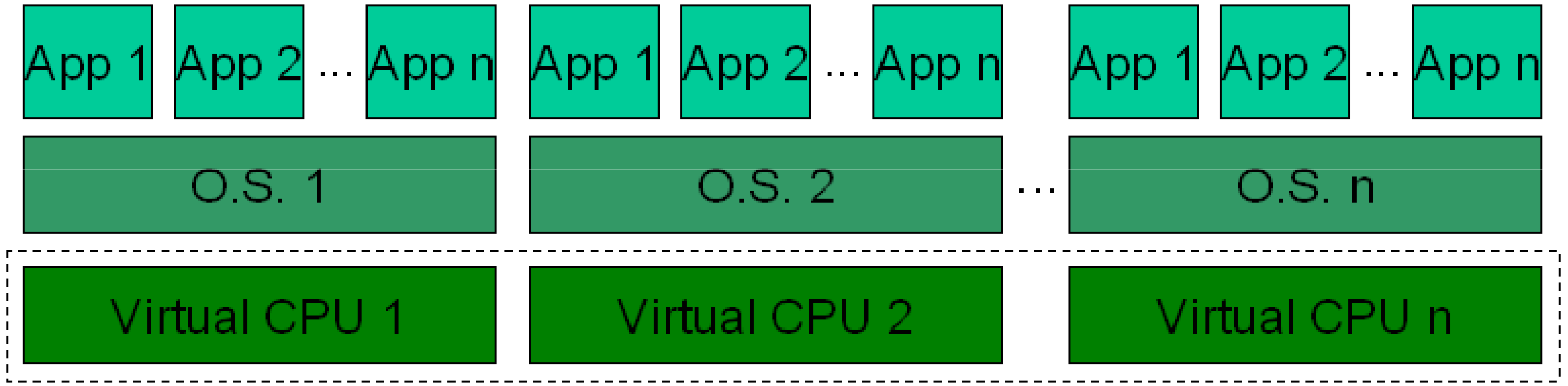
On multitasking, there is a single operating system and several programs running in parallel.

Hyper-Threading



Hyper-Threading (HT) simulates two CPUs where there is just one physical CPU for balancing performance using SMP (Symmetric Multi Processing), and these two CPUs cannot be used separately.

Virtualization



On virtualization, you can have several operating systems running in parallel, each one with several programs running. Each operating system runs on a “virtual CPU” or “virtual machine”.

If a CPU has both Hyper-Threading and Virtualization Technology each virtual CPU will appear to the operating system as if two CPUs are available on the system for symmetric multiprocessing.

Types of virtualization

- Client virtualization
- Network virtualization
- Storage virtualization
- Server virtualization

Client virtualization

- Virtualization capabilities residing on a client (a desktop or laptop PC).
- Types:
  - Application packaging: isolating an application that runs on a client machine from the underlying operating system.
    - Executing the application on top of a software product that gives each application its own virtual set of system resources — stuff like files and registry entries.
    - Bundling the application and the virtualization software into a single executable program that is downloaded or installed (*sandboxed*).
    - Examples: SVS from Altiris, Thinstall's Virtualization Suite, and Microsoft's SoftGrid.
  - Application streaming: storing the proper versions of applications on servers in the data center, and when an end user wants to use a particular application, it's downloaded on the fly to the end user's machine.
    - Examples: AppStream's Virtual Image Distribution, Softricity's Softgrid for Desktops, and Citrix's Presentation Server.
  - Hardware emulation: the virtualization software presents a software representation of the underlying hardware that an operating system would typically interact with.
    - Examples: VMware's VMware Server and Microsoft's Virtual Server

Network virtualization

- The process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network.

- Network virtualization is categorized as either *external*, combining many networks, or parts of networks, into a virtual unit, or *internal*, providing network-like functionality to the software containers on a single system. Whether virtualization is internal or external depends on the implementation provided by vendors that support the technology.

Storage virtualization

- Direct-Attached Storage (DAS)
  - Hard drives attached to whatever physical server running the application.
- Network-Attached Storage (NAS)
  - Is a machine that sits on the network and offers storage to other machines, special hardware appliances are used to fill with many hard drives and provide terabytes of storage.
  - NAS uses standard data communication protocols to send data back and forth between a server and the NAS device.
- Storage Area Network (SAN)
  - Acronym easy to confuse with a similar solution, i.e. NAS;
  - Servers use own SAN interface device called a Host Bus Adapter (HBA) to connect to their SAN.
  - SANs use a specialized network protocol---either something called Fibre Channel or the alternative choice, iSCSI (Internet Small Computer System Interface) for SAN network communication.
  - Fibre Channel SANs use specialized data communication protocols to send data back and forth between a server and the data storage device on a separate network, specialized hardware is needed on servers.
  - iSCSI-based SANs are able to use the standard corporate network for the data traffic, although this can raise the same kind of network congestion issues posed by NAS storage.

Popek and Goldberg virtualization requirements

- The Popek and Goldberg virtualization requirements are a set of conditions sufficient for a computer architecture to support system virtualization efficiently. They were introduced by Gerald J. Popek and Robert P. Goldberg in their 1974 article “Formal Requirements for Virtualizable Third Generation Architectures”.

- Three properties of interest when analyzing the environment created by a VMM:
  - Equivalence / Fidelity: A program running under the VMM should exhibit a behavior essentially identical to that demonstrated when running on an equivalent machine directly.
  - Resource control / Safety: The VMM must be in complete control of the virtualized resources.
  - Efficiency / Performance: A statistically dominant fraction of machine instructions must be executed without VMM intervention.

- Instruction Set Architecture (ISA)
  - Privileged instructions : Those that trap if the processor is in user mode and do not trap if it is in system mode.
  - Control sensitive instructions: Those that attempt to change the configuration of resources in the system.
  - Behavior sensitive instructions: Those whose behavior or result depends on the configuration of resources (the content of the relocation register or the processor's mode).

- Theorem 1.
  - For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.

Popek and Goldberg virtualization requirements (Cont.d)

- Theorem 2.
  - A conventional third-generation computer is recursively virtualizable if:
    - 1. it is virtualizable and
    - 2. a VMM without any timing dependencies can be constructed for it.
  - Some architectures, like the non-hardware-assisted x86 do not meet these conditions, so they cannot be virtualized in the classic way. But architectures can still be fully virtualized (in the x86 case meaning at the CPU and MMU level) by using different techniques like binary translation, which replaces the sensitive instructions that do not generate traps, which are sometimes called critical instructions.
- Handling critical instructions:
  - The conditions for ISA virtualization expressed in Theorem 1 may be relaxed at the expense of the efficiency property. VMMs for non-virtualizable ISAs (in the Popek and Goldberg's sense) have routinely been built.
  - The virtualization of such architectures requires correct handling of critical instructions, i.e., sensitive but unprivileged instructions. One approach, known as patching, adopts techniques commonly used in dynamic recompilation: critical instructions are discovered at run-time and replaced with a trap into the VMM. Various mechanisms, such as the caching of emulation code or hardware assists, have been proposed to make the patching process more efficient. A different approach is that of paravirtualization, which requires guest operating systems to be modified (ported) before running in the virtual environment.

Server virtualization

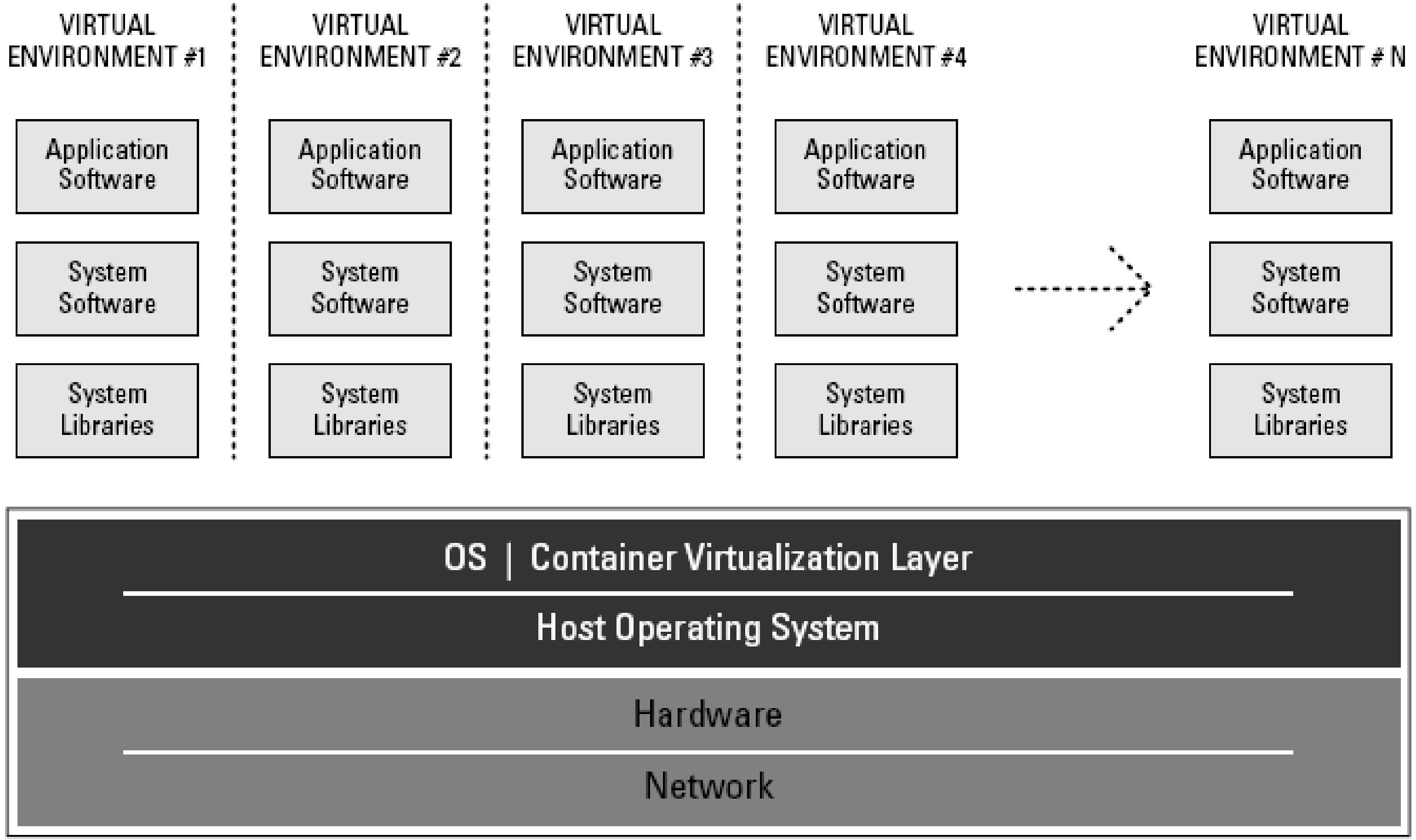
- Operating system virtualization (containers)
- Hardware emulation (full virtualization)
- Paravirtualization
  - Doesn’t create an entire virtual machine to host the guest OS, but rather enables the guest OS to interact directly with the hypervisor.
  - The “para” in paravirtualization doesn’t really mean anything. It’s actually just a smart-alec term used to sneer at hardware emulation virtualization.
  - Examples: Xen.

Source: Virtualization for dummies

Operating system virtualization (containers)

- Runs on top of an existing host operating system and provides a set of libraries that applications interact with, giving each application the illusion that it is running on a machine dedicated to its use.
- Typically limits operating system choice.
- Examples: Sun (as part of the Solaris operating system) and SWsoft.

Operating system virtualization

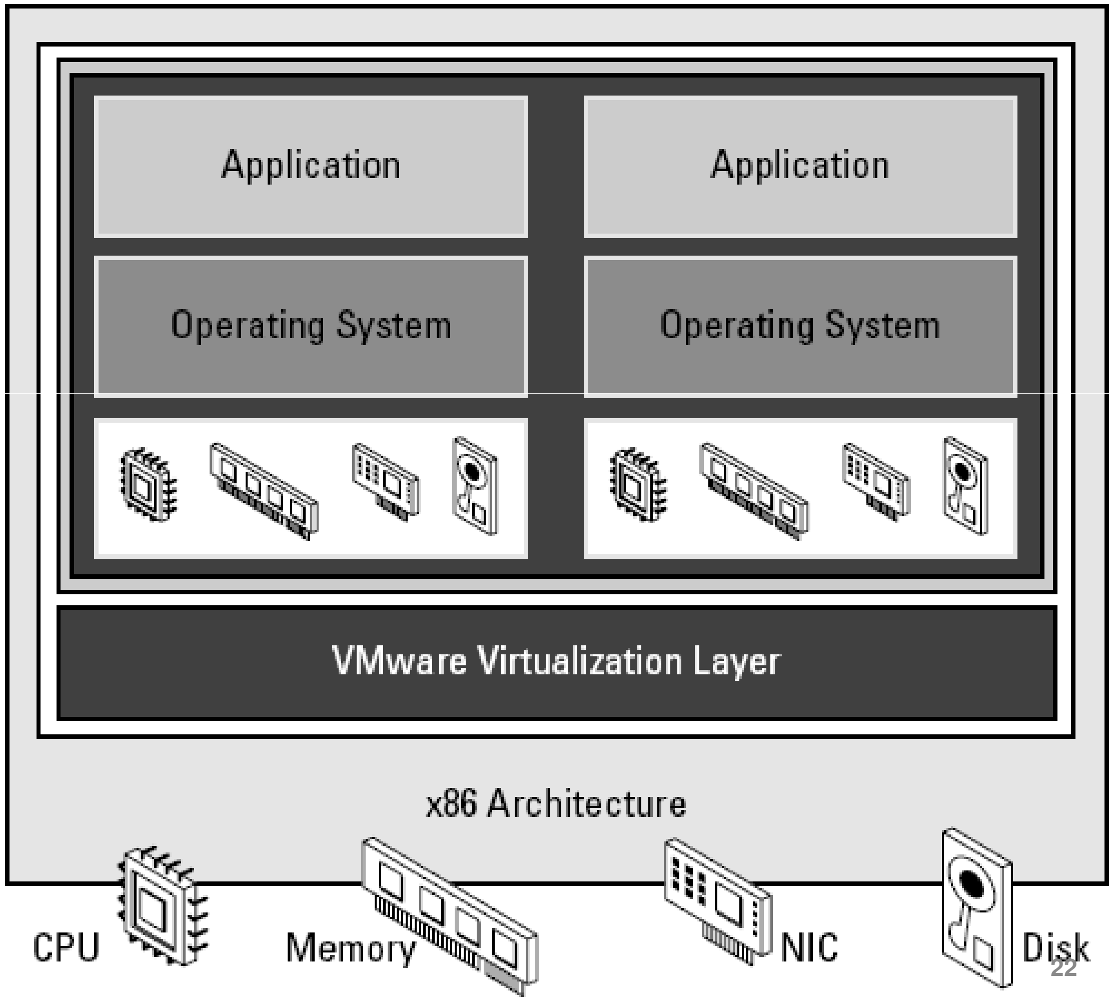


Aalto University

Hardware emulation (full virtualization)

- The virtualization software (usually referred to as a hypervisor) presents an emulated hardware environment that guest operating systems operate upon.
- Performs the guest OS modifications at runtime through a technique known as Binary Translation; essentially, when the guest OS is installed, the virtualization software cleverly rearranges the internal guest OS software so that the calls that originally attempted to access physical hardware resources are now directed to resources within the VMM.
- Drawback: Performance penalty; Device driver availability; Device driver inflexibility.
- Examples: VMware Server and ESX Server; Microsoft's Virtual Server.

Hardware emulation (full virtualization)



Full Virtualization

- A virtualization technique used to provide a certain kind of virtual machine environment, namely, one that is a complete simulation of the underlying hardware. Full virtualization requires that every salient feature of the hardware be reflected into one of several virtual machines – including the full instruction set, input/output operations, interrupts, memory access, and whatever other elements are used by the software that runs on the bare machine, and that is intended to run in a virtual machine.
- An important example of full virtualization was that provided by the control program of IBM's CP/CMS operating system. It was first demonstrated with IBM's CP-40 research system in 1967, then distributed via open source in CP/CMS in 1967-1972, and re-implemented in IBM's VM family from 1972 to the present.
- This simulation was comprehensive, and was based on the Principles of Operation manual for the hardware. It thus included such elements as instruction set, main memory, interrupts, exceptions, and device access. The result was a single machine that could be multiplexed among many users.
- Full virtualization is only possible given the right combination of hardware and software elements.
- Similarly, full virtualization was not quite possible with the x86 platform until the 2005-2006 addition of the AMD-V and Intel VT-x extensions.
- VMware, for instance, employs a technique called binary translation to automatically modify x86 software on-the-fly to replace instructions that "pierce the virtual machine" with a different, virtual machine safe sequence of instructions; this technique provides the appearance of full virtualization.

Full Virtualization (Cont.d)

- A key challenge for full virtualization is the interception and simulation of privileged operations, such as I/O instructions. The effects of every operation performed within a given virtual machine must be kept within that virtual machine – virtual operations cannot be allowed to alter the state of any other virtual machine, the control program, or the hardware.

- Some machine instructions can be executed directly by the hardware, since their effects are entirely contained within the elements managed by the control program, such as memory locations and arithmetic registers. But other instructions that would "pierce the virtual machine" cannot be allowed to execute directly; they must instead be trapped and simulated. Such instructions either access or affect state information that is outside the virtual machine.

- Full virtualization has proven highly successful for
  - a) sharing a computer system among multiple users,
  - b) isolating users from each other (and from the control program) ,
  - c) emulating new hardware to achieve improved reliability, security and productivity.

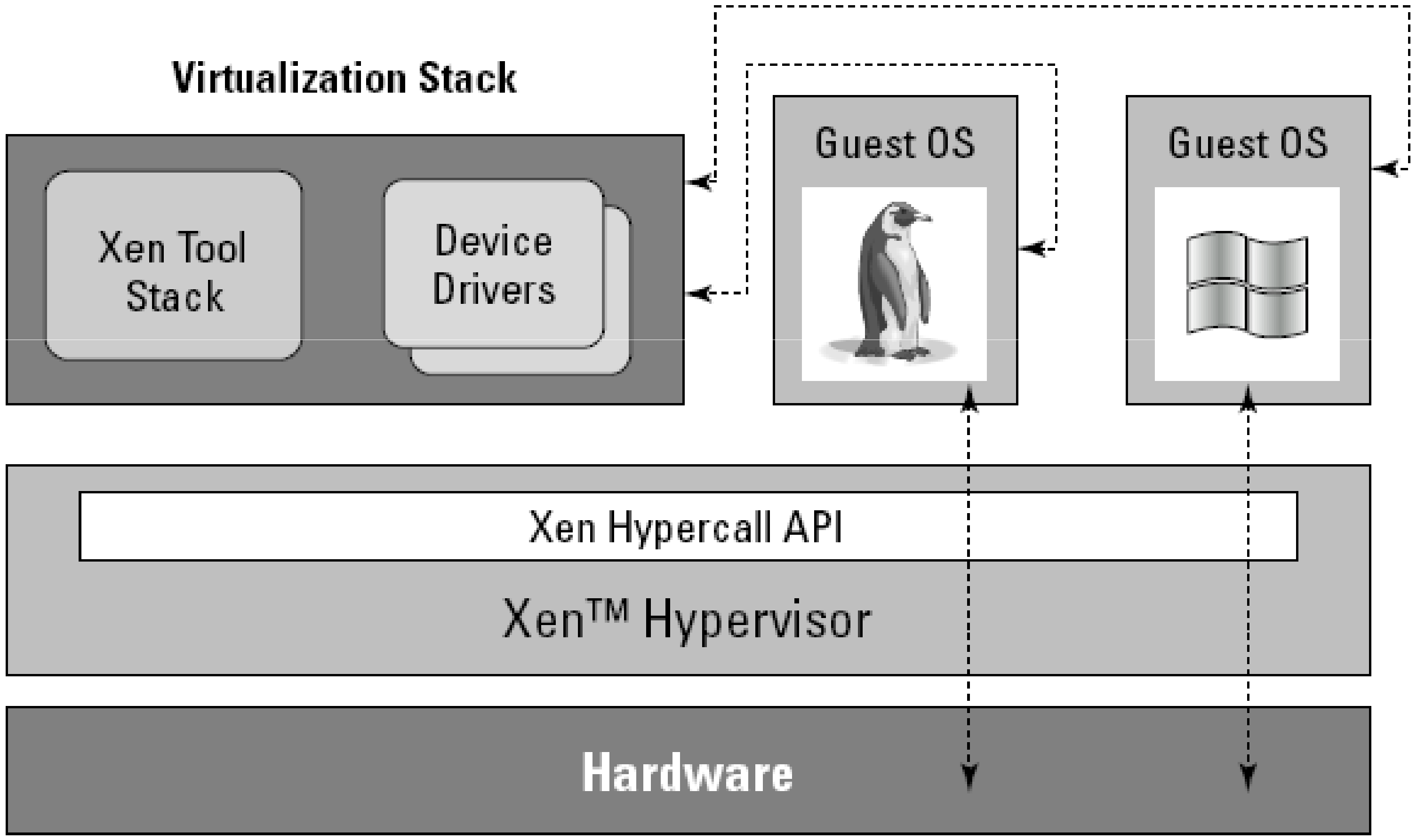
Paravirtualization

- Doesn’t create an entire virtual machine to host the guest OS, but rather enables the guest OS to interact directly with the hypervisor.
- The “para” in paravirtualization doesn’t really mean anything. It’s actually just a smart-alec term used to sneer at hardware emulation virtualization.
- Examples: Xen.

Definition

- Paravirtualization is a virtualization technique that presents a software interface to virtual machines that is similar but not identical to that of the underlying hardware.
- A successful paravirtualized platform may allow the virtual machine monitor (VMM) to be simpler (by relocating execution of critical tasks from the virtual domain to the host domain), and/or reduce the overall performance degradation of machine-execution inside the virtual-guest.
- Paravirtualization requires the guest operating system to be explicitly ported for the para-API — a conventional OS distribution which is not paravirtualization-aware cannot be run on top of a paravirtualized VMM.
- A hypervisor provides the virtualization abstraction of the underlying computer system. In full virtualization, a guest operating system runs unmodified on a hypervisor. However, improved performance and efficiency is achieved by having the guest operating system communicate with the hypervisor. By allowing the guest operating system to indicate its intent to the hypervisor, each can cooperate to obtain better performance when running in a virtual machine. This type of communication is referred to as paravirtualization.

Paravirtualization



Paravirtualization vs. Binary Translation (BT)

- BT changes "critical" or "dangerous" code into harmless code on the fly;
- Paravirtualization does the same thing in the source code.
- Changing the source code allows more flexibility than changing everything on the fly.
- The hypervisor provides hypercall interfaces for critical kernel operations such as memory management, interrupt handling, and time keeping. These hypercalls will only happen when it is necessary. For example, most of the memory management is done by the different guest OSes. The Hypervisor will only be "called" for things like page table updates and DMA (Direct Memory Access) accesses.

Hardware-assisted virtualization

- The hardware provides architectural support that facilitates building a virtual machine monitor and allows guest OSes to be run in isolation.

- Hardware-assisted virtualization was first introduced on the IBM System/370 in 1972, for use with VM/370, the first virtual machine operating system. In 2005 and 2006, Intel and AMD provided additional hardware to support virtualization, VT-x and AMD-V respectively.

- Intel® Virtualization Technology for servers. http://www.intel.com/technology/virtualization/demos/closer/demo.htm
- http://www.youtube.com/watch?v=gqZrarZiHp8

Why hardware-assisted?

- Virtualization Technology uses the same idea of Virtual 8086 (V86) mode, which is available since 386's. With V86 mode you can create several virtual 8086 machines to run DOS-based programs in parallel. With VT you can create several "complete" virtual machines to run full operating systems in parallel.

- But if there are software like VMware that enables virtualization, why implement Virtualization Technology inside the CPU? The advantage is that CPUs with Virtualization Technology have some new instructions to control virtualization. With them, controlling software (called VMM, Virtual Machine Monitor) can be simpler, thus improving performance compared to software-only solutions.

How it works?

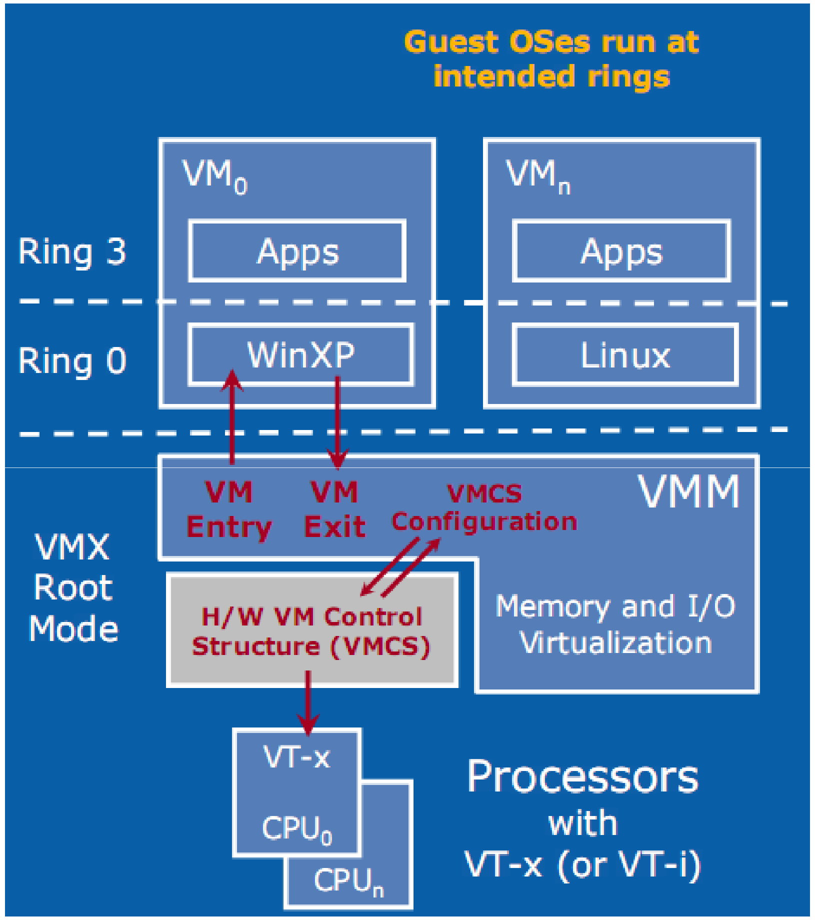
- Processors with Virtualization Technology have an extra instruction set called Virtual Machine Extensions or VMX. VMX brings 10 new virtualization-specific instructions to the CPU: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMWRITE, VMCALL, VMLAUCH, VMRESUME, VMXOFF and VMXON.

- There are two modes to run under virtualization: root operation and non-root operation. Usually only the virtualization controlling software, called Virtual Machine Monitor (VMM), runs under root operation, while operating systems running on top of the virtual machines run under non-root operation. Software running on top of virtual machines is also called “guest software”.

- To enter virtualization mode, the software should execute the VMXON instruction and then call the VMM software. Then VMM software can enter each virtual machine using the VMLAUNCH instruction, and exit it by using the VMRESUME. If VMM wants to shutdown and exit virtualization mode, it executes the VMXOFF instruction.

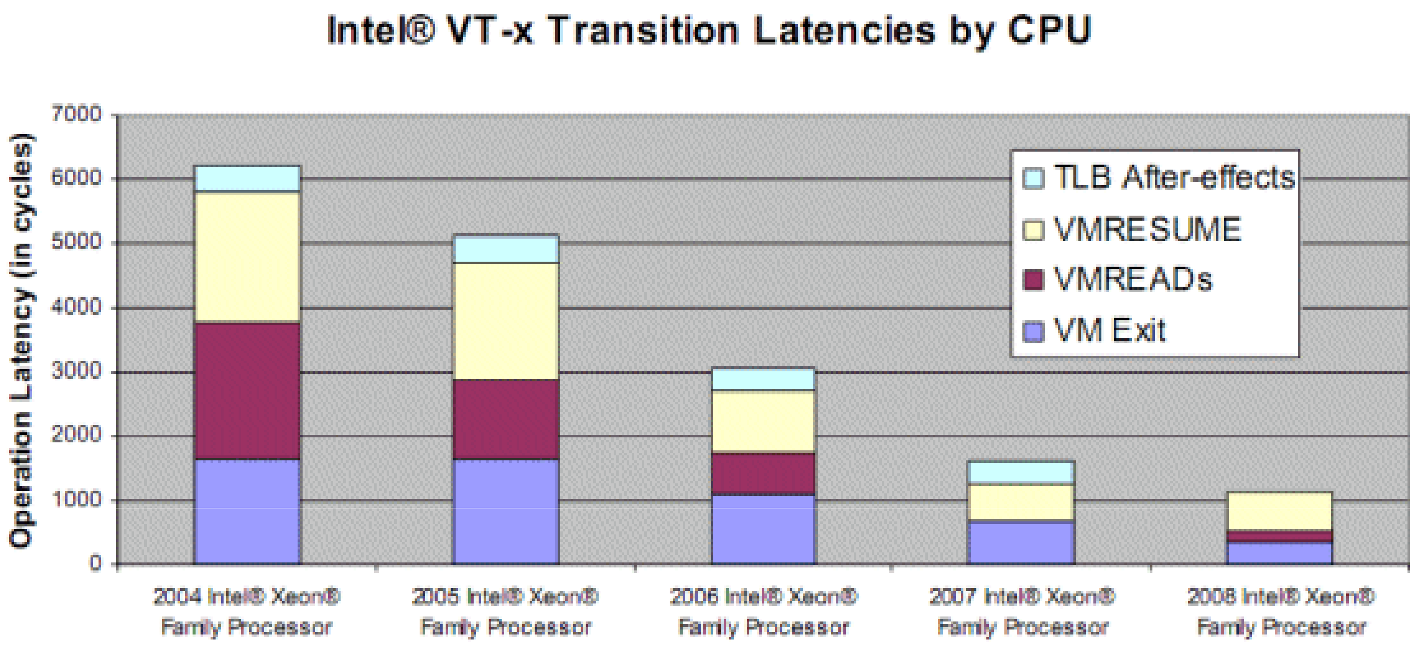
H-W virtualization

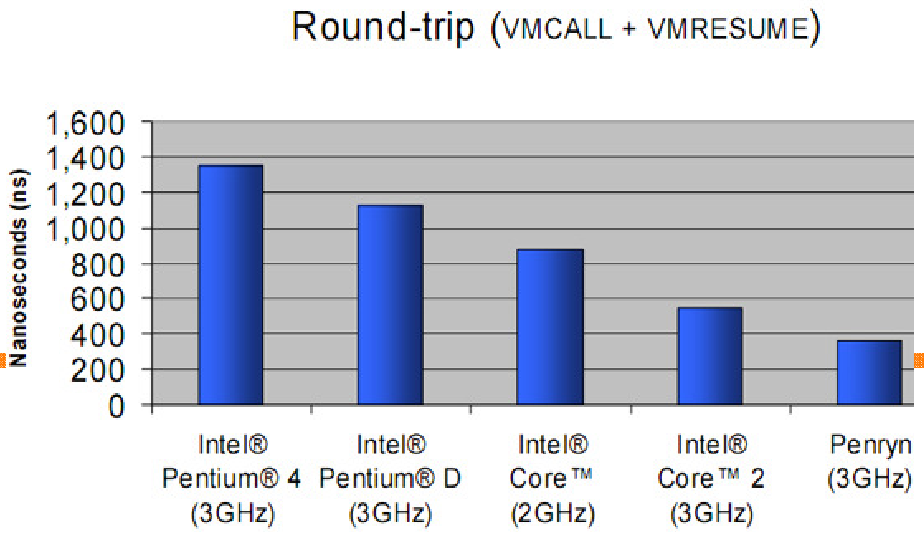
- Guest OS runs at its intended privilege level (ring 0), and the VMM is running at a new ring with an even higher privilege level (Ring -1, or "Root mode").
- Advantage: System calls do not necessarily result in VMM interventions: as long as system calls do not involve critical instructions, the guest OS can provide kernel services to the user applications.


Guest OSes run at intended rings
VM0
VMn
Ring 3
Apps
Apps
Ring 0
WinXP
Linux
VMX Root Mode
VM Entry
VM Exit
VMCS Configuration
VMM
H/W VM Control Structure (VMCS)
Memory and I/O Virtualization
VT-x
CPU0
CPUn
Processors with VT-x (or VT-i)

Source: http://www.anandtech.com/show/2480/9

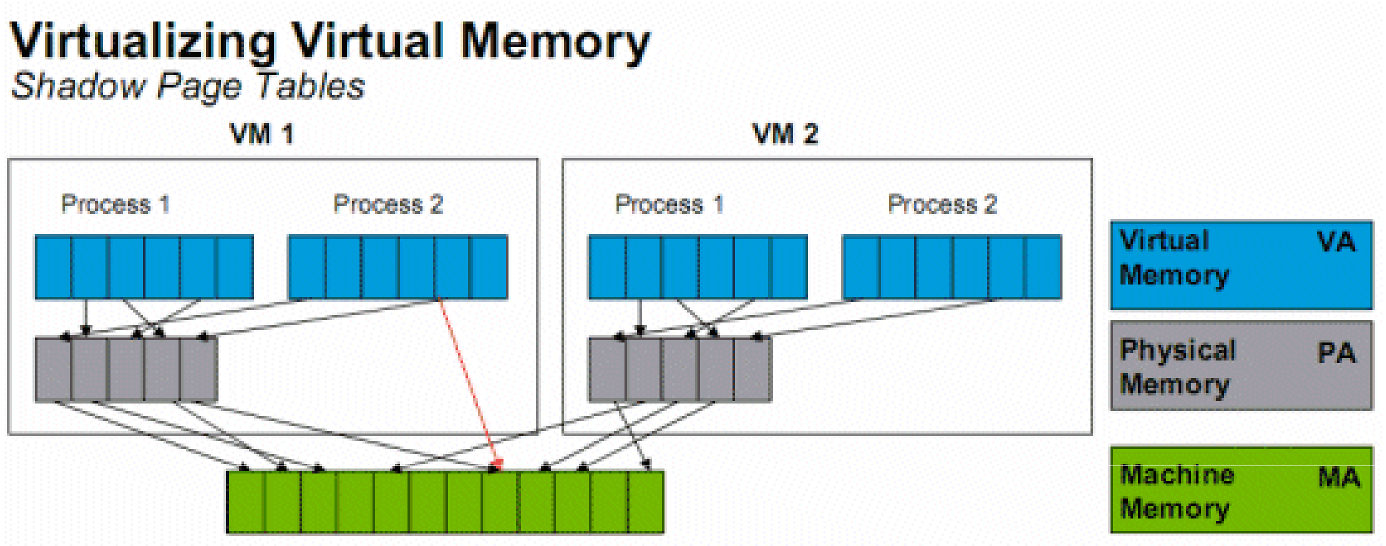
Performance





Aalto University

Shadow page table



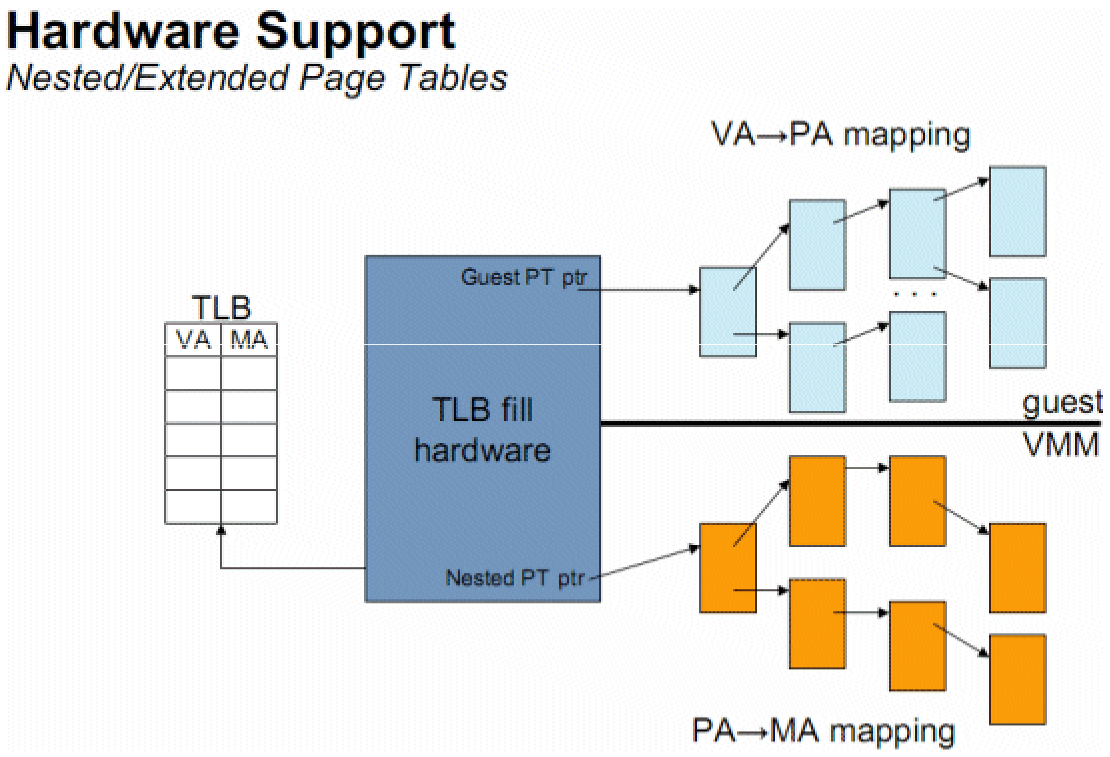
• Without shadow pages, we would have to translate virtual memory (blue) into "guest OS physical memory" (gray) and then translate the latter into the real physical memory (green);

• The shadow page table avoids the double bookkeeping by making the MMU work with a virtual memory (of the guest OS, blue) to real physical memory (green) page table, effectively skipping the intermediate "guest OS physical memory" step;

• But each update of the guest OS page tables requires some "shadow page table" bookkeeping, which is costly.

Source: http://www.anandtech.com/show/2480/10

Nested/Extended Page Table

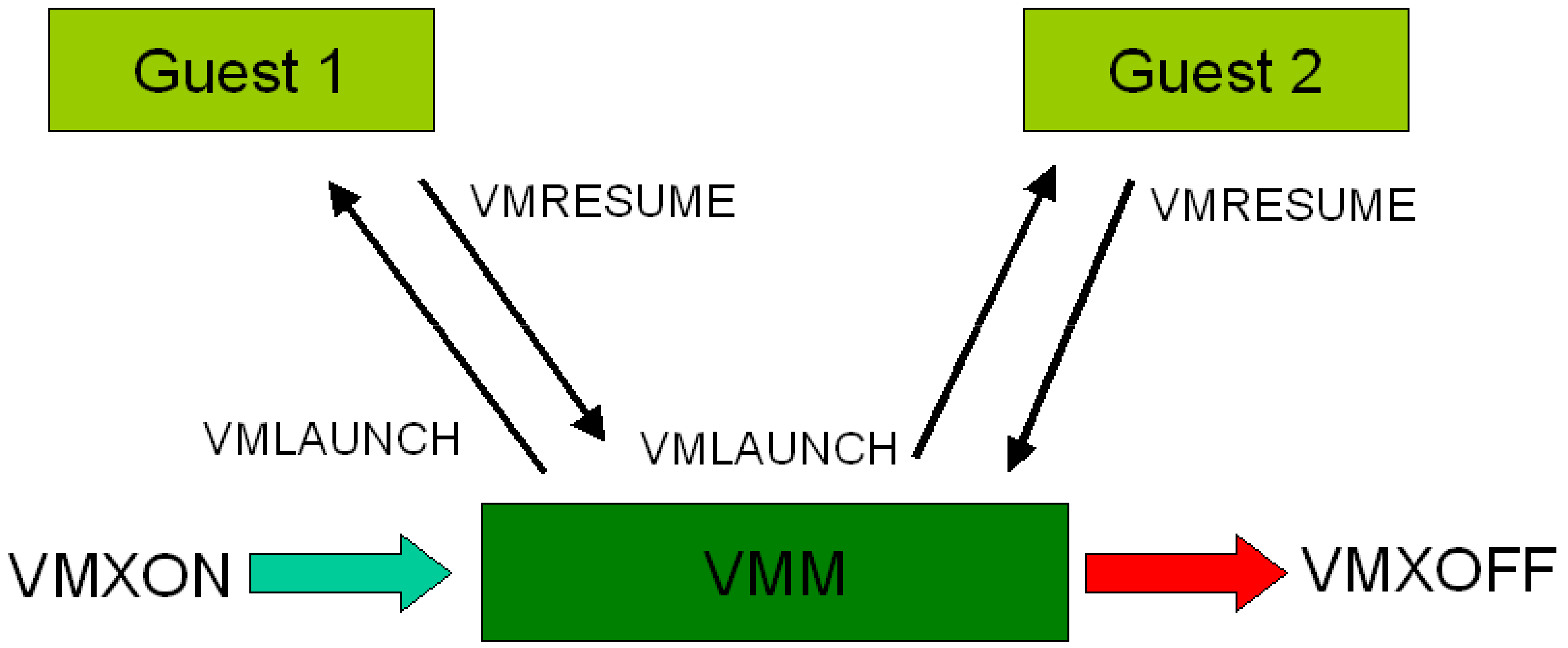


Conclusion of hardware virtualization

- First generation hardware virtualization had the performance downside, but enabled many steps forward. It has been improved, and is now adopted by VMware to support 64-bit guest OSes and by Xen to run unmodified OSes (such as Windows).
- Second generation virtualization (VT-x+EPT and AMD-V+NPT) is more promising, but while it can improve performance significantly it is not guaranteed that it will improve performance across all applications due to the heavy TLB miss cost.
- VMware ESX is the best example of a hybrid model. ESX uses paravirtualized drivers for the most critical I/O components (but not for the CPU), uses emulation for the less important I/O, Binary Translation to avoid the high "trap and emulate" performance penalty, and hardware virtualization for 64-bit guests. In this way, virtualized applications perform quite well, in some cases almost as if there is no extra layer (the VMM).

Source: http://www.anandtech.com/show/2480/13

Virtualization technology operation



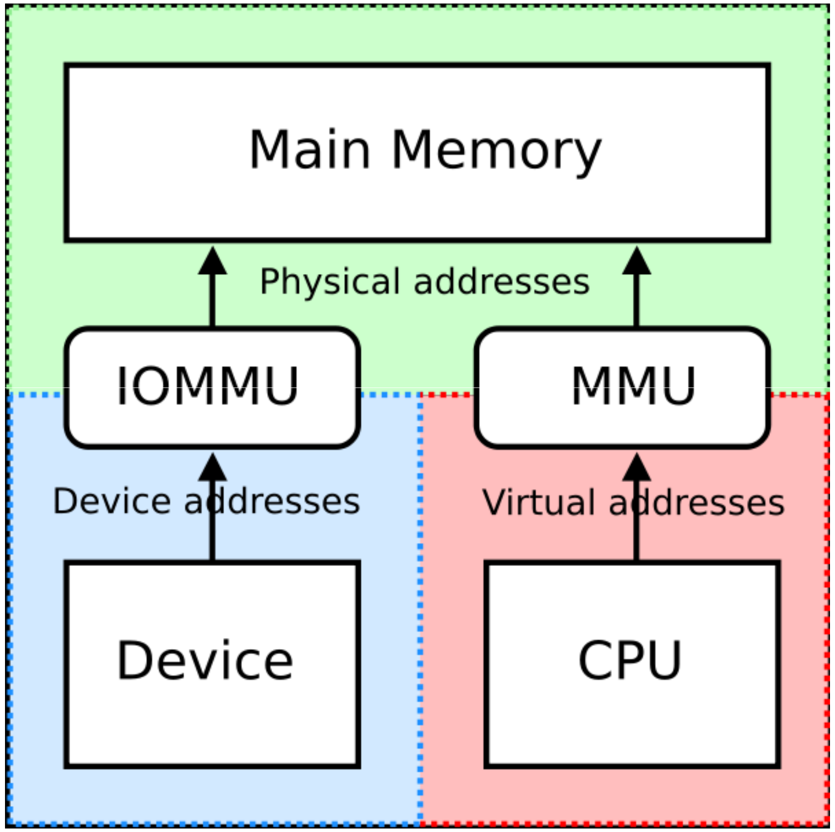
Each guest shown in figure above can be a different operating system, running its own software (even several programs at the same time).

IOMMU

- In computing, an input/output memory management unit (IOMMU) is a memory management unit (MMU) that connects a DMA-capable I/O bus to the main memory.

- Like a traditional MMU, which translates CPU-visible virtual addresses to physical addresses, the IOMMU takes care of mapping device-visible virtual addresses (also called device addresses or I/O addresses in this context) to physical addresses. Some units also provide memory protection from misbehaving devices.

IOMMU (Cont.d)



Comparison of IOMMU to MMU

References

- Virtualization for dummies. Bernard Golden. Wiley Publishing, Inc. ISBN: 978-0-470-14831-0.
- Wikipedia.