Introducción a la Virtualización

Virtualización en el Intel x86

v.20201117

Arquitecturas Virtuales

Depto. de Arquitectura de Computadores
Universidad de Málaga

El resurgir de la virtualización

- Años 70: uso de las primeras máquinas virtuales
  - 1972: IBM VM/370 (puede ejecutar incluso una MV en otra MV)
- Años 90:
  - Intel x86 se convierte en una arquitectura predominante
  - Los "*commodity OSs*" carecen de prestaciones empresariales
- ~1999: VMware pionera en virtualización de plataformas basadas en x86
- ~2006: Soporte HW en CPUs de Intel y AMD

Fuente: A Comparison of SW & HW techniques for x86 virtualizacion.
K. Adams y O. Agenson, 2006

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Arquitectura Intel x86

Generation	First introduced	Prominent Consumer CPU brands	linear / physical address space	Notable (new) features
1 **(IA-16)**	1978	**Intel 8086**, Intel 8088	**16-bit** / 20-bit (*segmented*)	first x86 microprocessors
2	1982	Intel 80186, Intel 80188, NEC V20		hardware for fast address calculations, fast mul/div etc
		Intel 80286	**16-bit** (30-bit *virtual*) / 24-bit (*segmented*)	MMU, for **protected mode** and a larger address space
3 **(IA-32)**	1985	**Intel 80386**, AMD Am386	**32-bit** (46-bit *virtual*) / 32-bit	**32-bit instruction set, MMU with paging**
4	1989	Intel 486		risc-like pipelining, integrated FPU, on-chip cache
5	1993	Pentium, Pentium MMX		superscalar, 64-bit databus, faster FPU, MMX
5/6	1996	Cyrix 6x86, Cyrix MII		register renaming, speculative execution

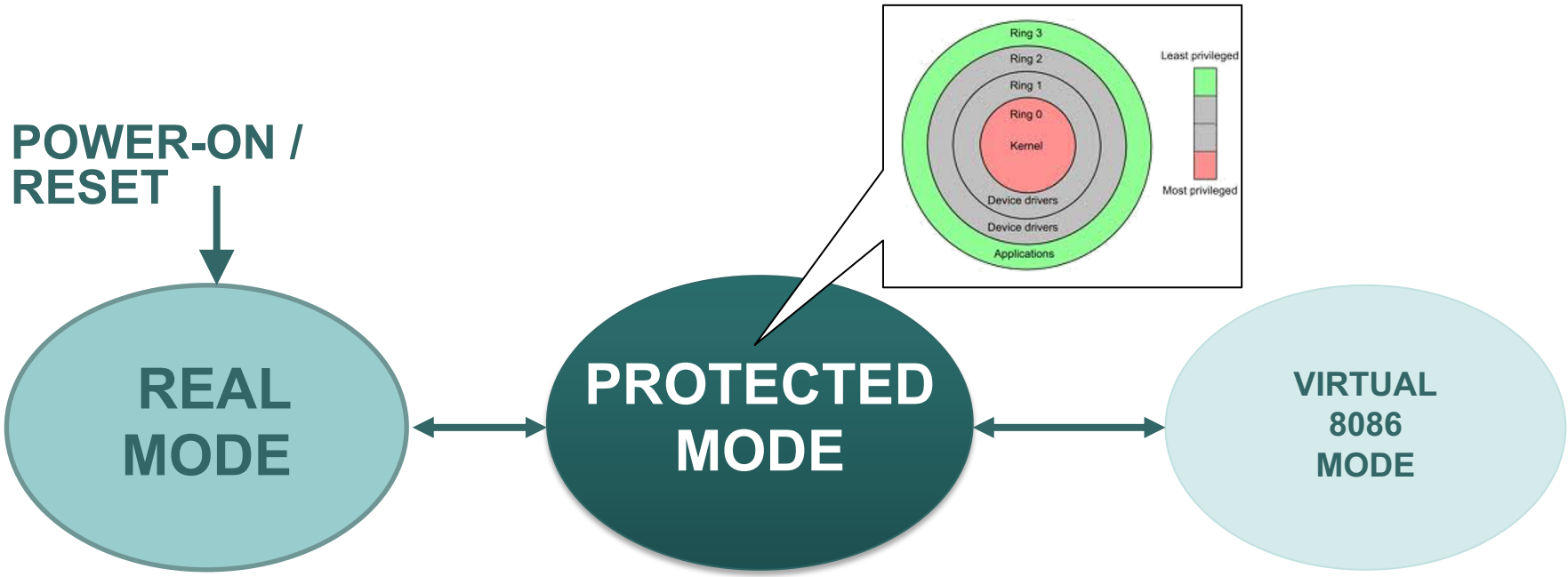
Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Arquitectura Intel x86

Generation	First introduced	Prominent Consumer CPU brands	linear / physical address space	Notable (new) features
6	1995	Pentium Pro, AMD K5, Nx586 (1994)	*as above* / **36**-bit *physical* (PAE)	u-op translation, PAE (not K5, Nx586), integrated L2 cache (not K5, Nx586)
	1997	AMD K6/-2/3, Pentium II/III		L3-cache support, 3D Now, SSE
7	1999	Athlon, Athlon XP		superscalar FPU, wide design (*up to three x86 instr./clock*)
	2000	Pentium 4		deeply pipelined, high frequency, SSE2, hyper-threading
8 (**x86-64**)	2003	Athlon 64, Opteron	**64-bit** / 40-bit *physical in first impl.*	**x86-64 instruction set**, on-die memory controller, hypertransport
	2004	Pentium 4 Prescott		very deeply pipelined, very high frequency, SSE3, **Intel VT-x (2005)**
9	2006	Intel Core 2		low power, multi-core, lower clock frequency, SSE4 (Penryn)
10	2007,08	AMD Phenom, Phenom II, Intel Core i3, i5,i7	*as above / 44-bit physical for Beckton Core i7*	monolithic quad-core, 128 bit FPUs, SSE4a, HyperTransport 3 or QuickPath, native memory controller, on-die L3 cache, modular design
11	2010	Intel Sandy Bridge, AMD Bulldozer		SSE5/AVX, highly modular design

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Modos de ejecución IA-32



Modo protegido

Es el modo de funcionamiento normal del procesador, en el que están disponibles todas sus características.

Modo (de direccionamiento) real

Es un modo de compatibilidad con el 8086, salvo por la capacidad de cambiar a modo protegido.

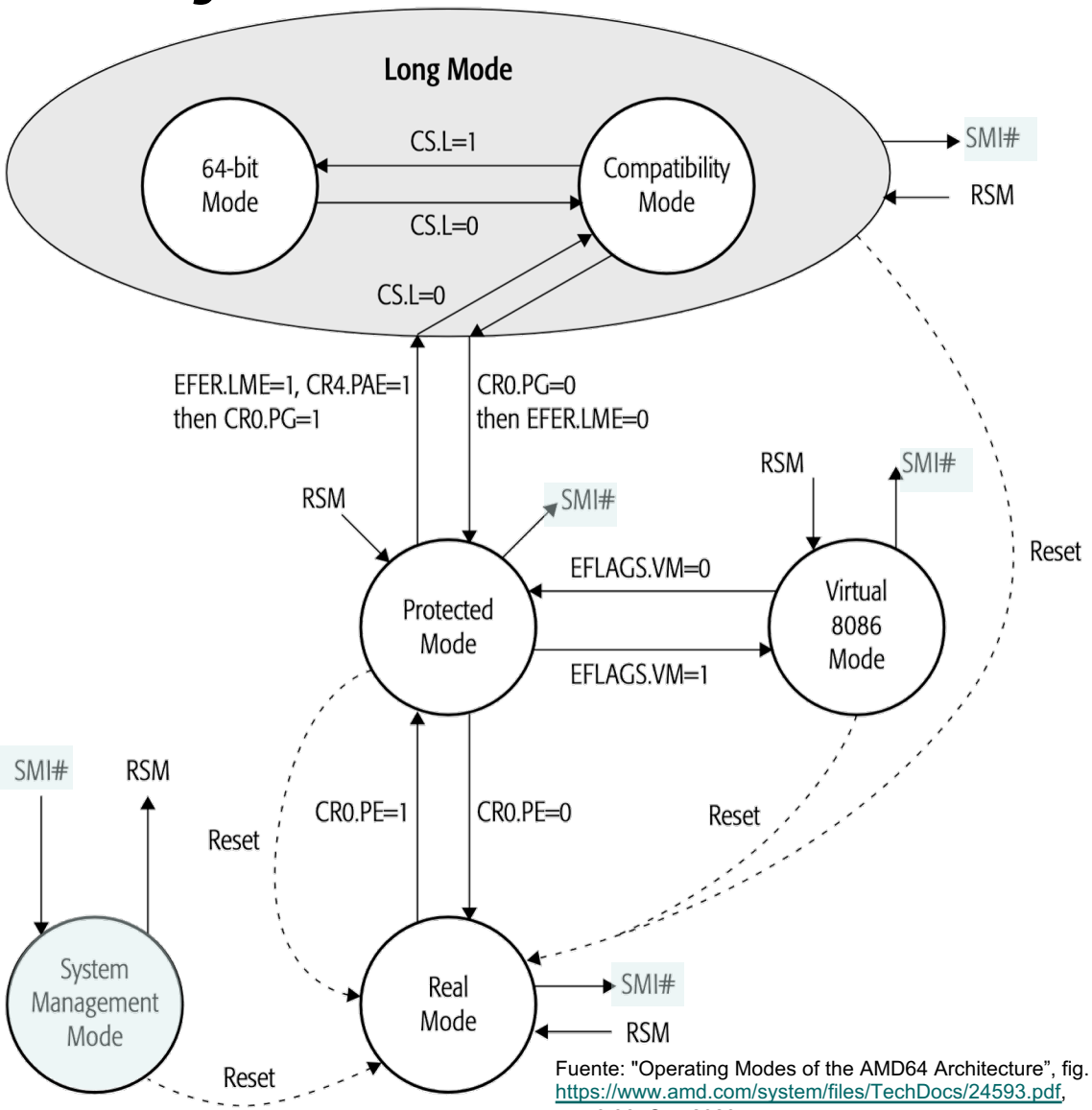
El procesador siempre comienza su ejecución en este modo (tras el arranque o un RESET).

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Modos de ejecución x86-64

System Management Mode (SMM)

- conocido como *ring "-2"*
- implica la suspensión del cualquier modo normal
- es similar a una interrupción no enmascarable (#SMI)
- disparado por eventos del hardware
- ejecuta una rutina en el hardware del fabricante
- máximo nivel de privilegio
- instrucción especial para salir de este modo (RSM = resume from SMM)



Fuente: "Operating Modes of the AMD64 Architecture", fig. 1.6, https://www.amd.com/system/files/TechDocs/24593.pdf, rev. 3.36, Oct. 2020

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

La arquitectura x86 “no es virtualizable” (en sentido estricto)

- Hay instrucciones “malas” 

  - instrucciones críticas que no causan *traps* cuando se ejecutan sin privilegio, ó
  - permiten a la MV “fisgonear” en el hipervisor
    - SGDT,SIDT,SLDT: Store Descriptor Table Registers
    - SMSW: Store Machine Status word
    - PUSHF(D),POPF(D): Access EFLAGS
    - LAR,LSL,VERR, VERW, POP, PUSH, CALL/JMP,INT, RET, STR, MOV: These instructions reference the privilege levels or can let a VM see the current privilege level without notifying the VMM:

- Estrategia: virtualizar el set de instrucciones IA-32 con traducción binaria *just-in-time*
  - *1998: VMWare U.S. Patent 6,397,242*

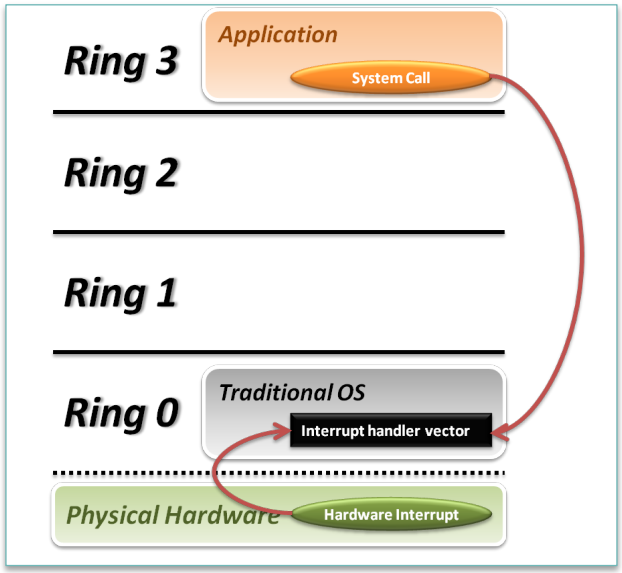
Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Soporte HW: Intel VT-x

- Para solventar estos problemas, Intel introduce un nuevo modo de operación en la arquitectura x86
- *VMX Root Mode*
  - Todas las instrucciones en este modo no son diferentes a las de la arquitectura tradicional
  - El software antiguo (*legacy*) debería de ejecutarse correctamente en este modo
  - El hipervisor debe correr en el modo privilegiado de este modo para tener control sobre todos los recursos
- *VMX Non-Root Mode*
  - Se rediseñan todas las instrucciones críticas (*sensitive*)
  - Las instrucciones críticas provocan un trap que pasa la CPU a modo *root*
  - El sistema operativo *guest* debe ejecutarse en este modo pudiendo ser virtualizado completamente mediante "*trap and emulate*".

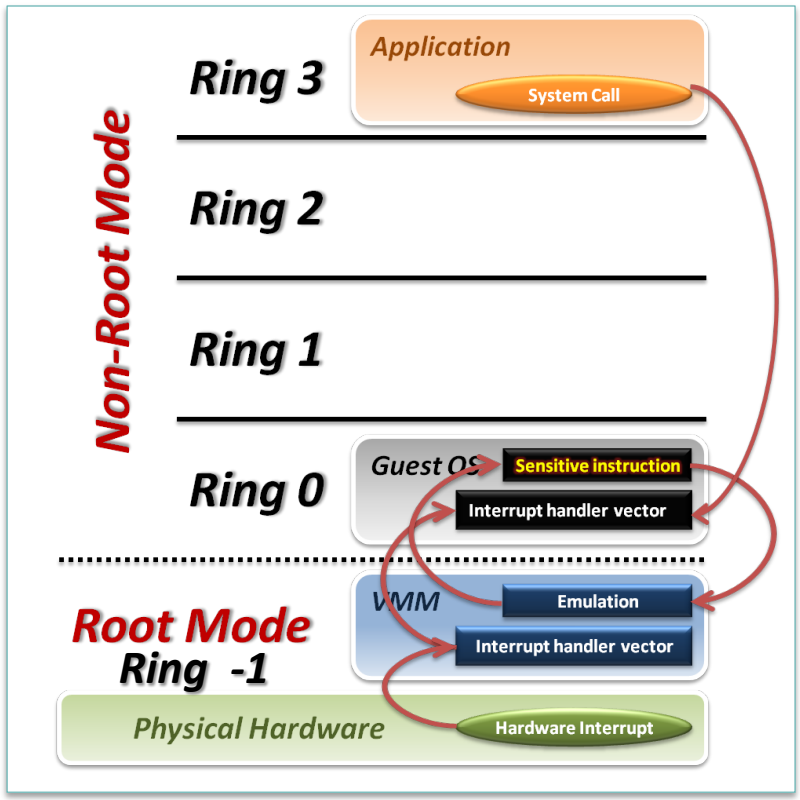
Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

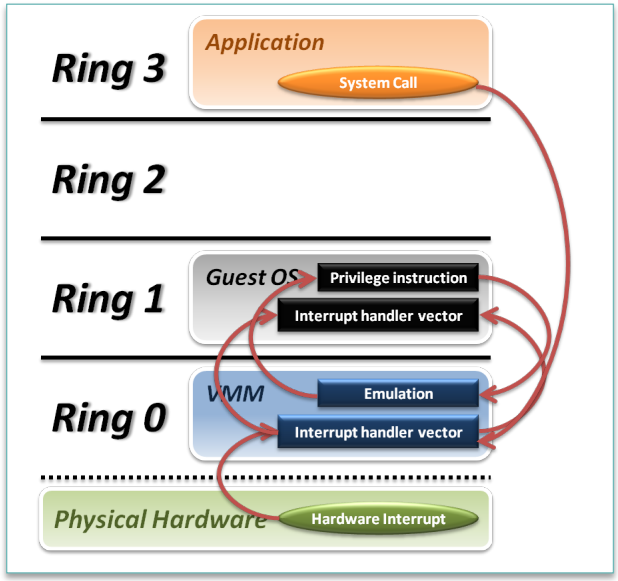
Soporte HW: Intel VT-x



S.O. tradicional

Con soporte HW





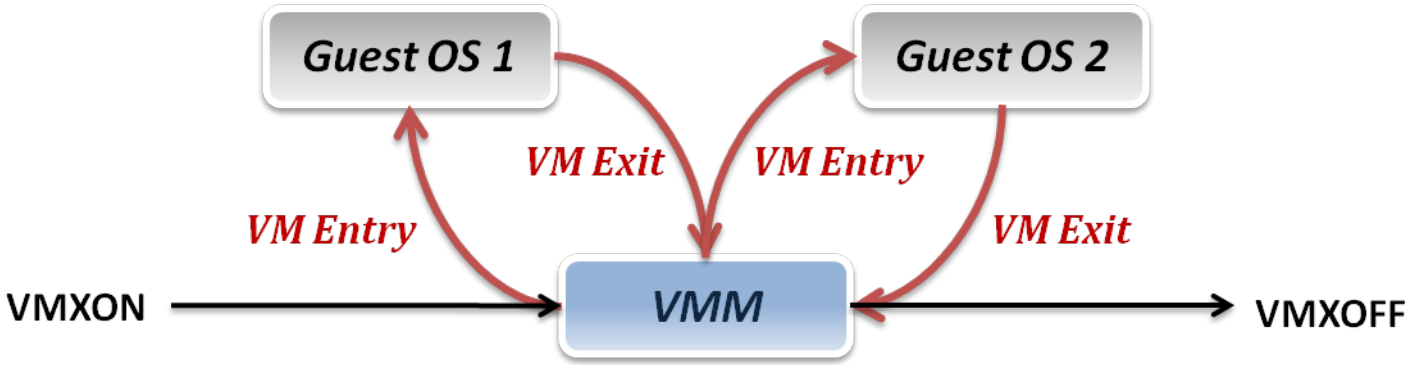
Modelo *Trap and emulate*

Ilustraciones: System Virtualization, Yeh-Ching Chung
http://www.cs.nthu.edu.tw/~ychung/syllabus/Virtualization.htm

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

MV (context) switch

- VMM switches different VMs with Intel VT-x :
  - **`VMXON/VMXOFF`**
    - These two instructions are used to turn on/off CPU Root Mode.
  - VM Entry
    - This is usually caused by the execution of `VMLAUNCH/VMRESUME` instructions, which will switch CPU mode from Root Mode to Non-Root Mode.
  - VM Exit
    - This may be caused by many reasons, such as hardware interrupts or sensitive instruction executions.
    - Switch CPU mode from Non-Root Mode to Root Mode.



Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

VT-x: *System State Management*

- Intel introduces a more efficient hardware approach for register switching, **VMCS** *(Virtual Machine Control Structure)* :
  - State Area
    - Store Host OS system state when VM-Entry.
    - Store Guest OS system state when VM-Exit.
  - Control Area
    - Control instruction behaviors in Non-Root Mode.
    - Control VM-Entry and VM-Exit process.
  - Exit Information
    - Provide the VM-Exit reason and some hardware information.
- Whenever VM Entry or VM Exit occur, CPU will automatically read or write corresponding information into VMCS.

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Soporte HW

- No sólo para la CPU, también para la memoria, red, E/S, …

	Solución Hardware	
	Intel	AMD
Instrucciones privilegiadas	VT-x	AMD-V
Virtualización de la RAM	EPT (Extended Page Tables)	NPT (Nested Paging)
Virtualización de la E/S	VT-d (Virtualization Technology for Directed I/O)	AMD-Vi (también llamado IOMMU)

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Virtualizadores para plataformas x86

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Un poco más de historia …

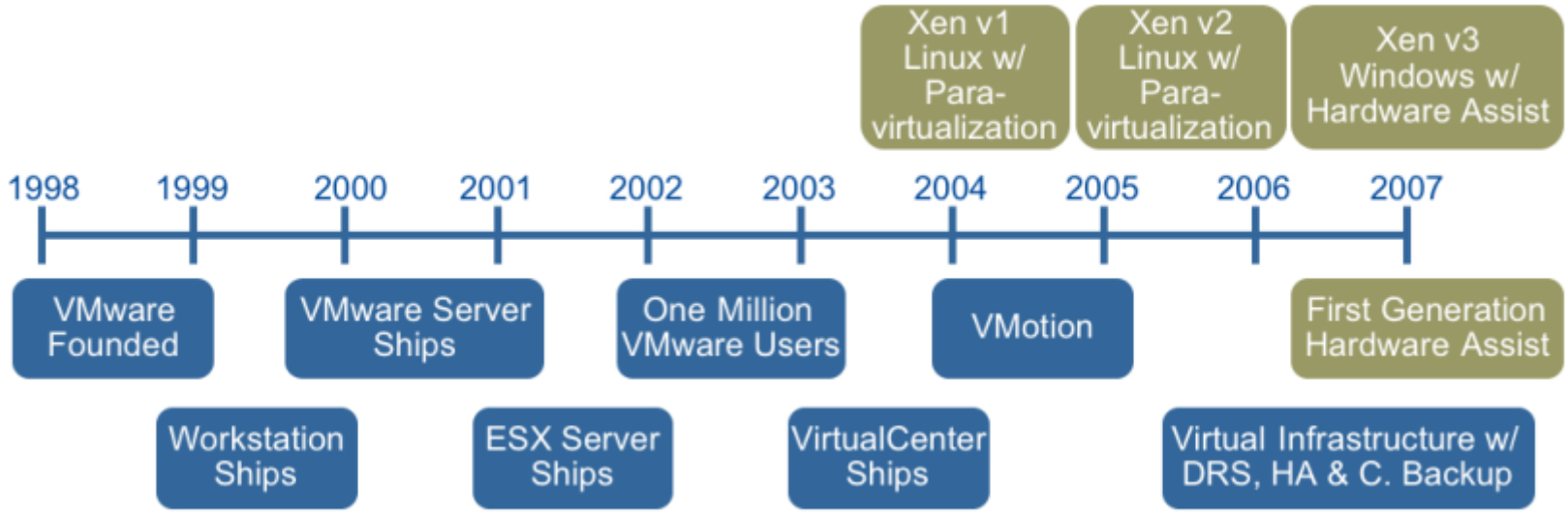


Ilustración: Understanding Full Virtualization, Paravirtualization, and Hardware Assist, 2008, Vmware Tech. Report

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

- Software de escritorio (hipervisores tipo 2)
  - Vmware workstation, player, fusion
- Sofware de servidor (hipervisores tipo 1)
  - VMware vSphere (ESXi)
- Muchos otros productos de virtualización dirigidos a infraestructuras virtuales:
  - vCloud,
  - Virtual desktops, …

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

KVM

- *Kernel-based Virtual Machine*
  - solución *open-source*
  - convierte Linux en un hipervisor
    - formado por módulos de kernel (uno para la infraestructura de virtualización, kvm.ko y otro dependiente de la CPU del host kvm-intel.ko)
  - Componentes:
    - Virtualization API (libvirt)
    - Managers: virsh (CLI), virt-manager (GUI)

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

QEMU

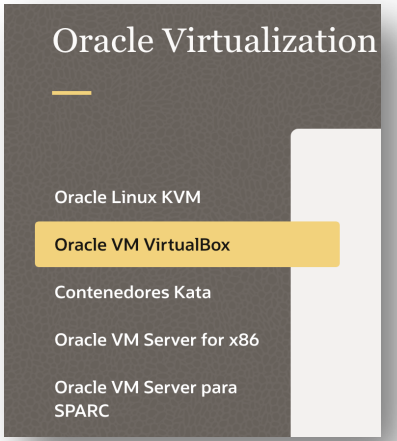
- QEMU = *Quick Emulator*
- Creado originalmente por Fabrice Bellard
- Considerado *"la navaja suiza"* de los virtualizadores
- *Free software* (GPL salvo algunas partes)
- Virtualización nativa tipo *hosted* (x86/x86-64) o bien emulación
- En linux, puede hacer uso de KVM (Kernel Virtual Machine) como acelerador (ambas plataformas, virtual y real, x86)
- Permite emular una gran variedad de plataformas y formatos:

  QEMU targets: x86_64, i386, arm, m68k, mips, ppc64, sparc64, cris, mips64, ppcemb, mips64el, mipsel, microblaze, ppc, sparc, …

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

- VirtualBox (Hosted hypervisor)

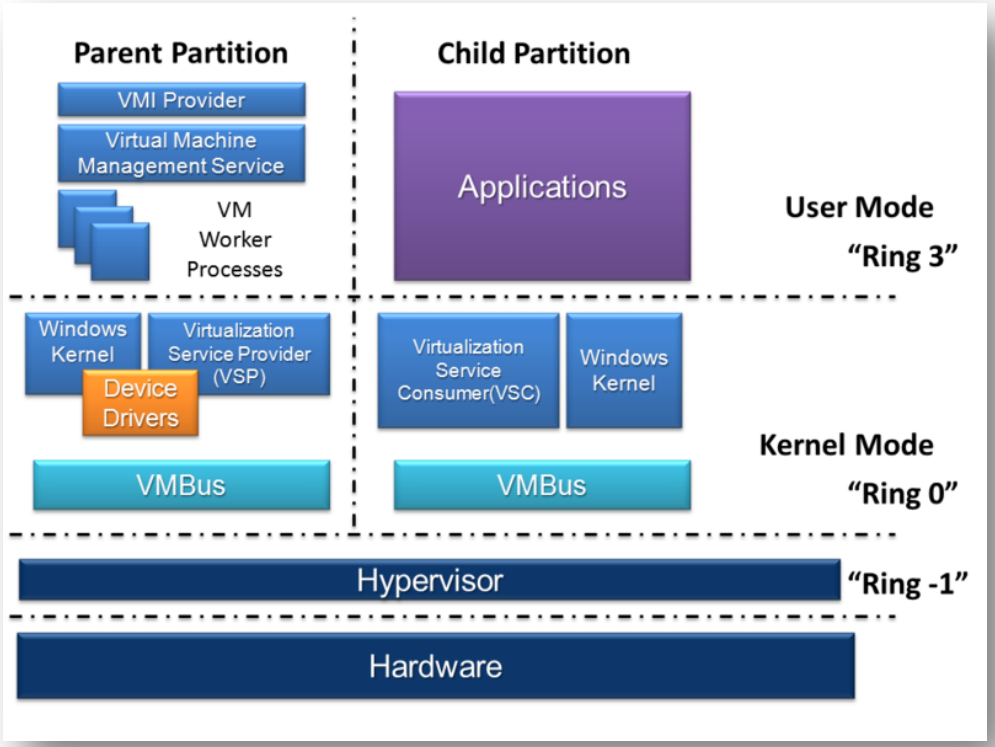
  - Software de virtualización para x86 / x86-64, originalmente de Sun
  - 2007: VirtualBox Open Edition, se ofrece como software abierto bajo GPL
  - Disponible en MS Windows, Linux, Solaris, Mac OS X y alguna otra plataforma
  - Compatibilidad con discos imágenes de VMware (.vmdk) y MS VirtualPC



- Oracle VM Server (Bare metal)
  - Instalable en servidores x86 y SPARC

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

- Hiper-V: orientado a servidores (hipervisor *bare metal* con maquina privilegiada)



Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

- Orígenes: proyecto de la universidad de Cambridge a finales de los 90
  - 2002: se convierte en un proyecto de código abierto
  - 2004: se funda XenSource, pero el proyecto *open source* sigue abierto
  - 2007: Citrix adquiere XenSource
- Hipervisor *bare metal* con máquina privilegiada

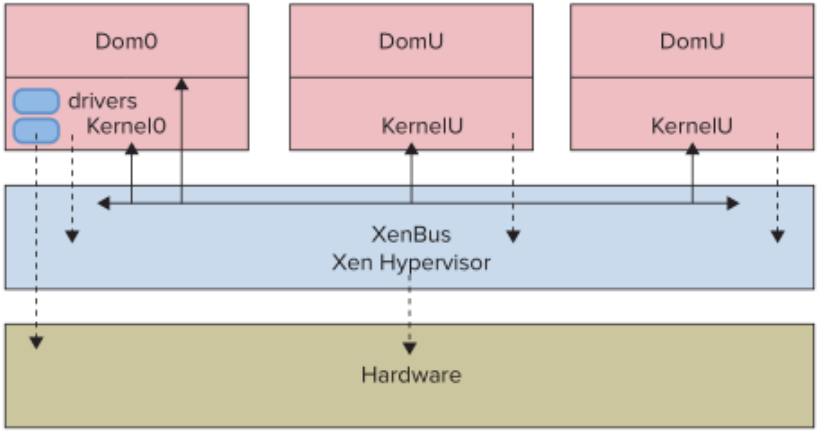


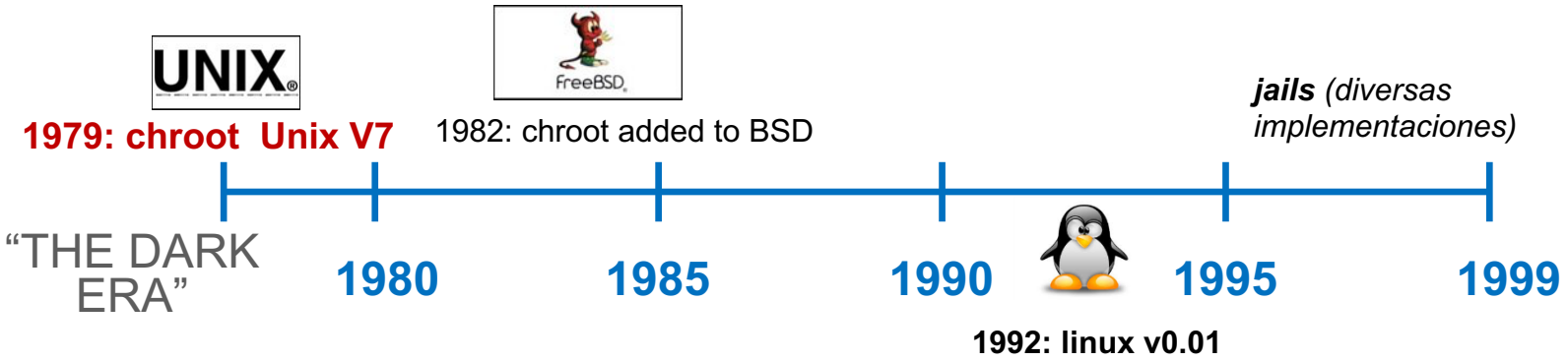
Ilustración: Virtualization essentials. M. Portnoy. 2012

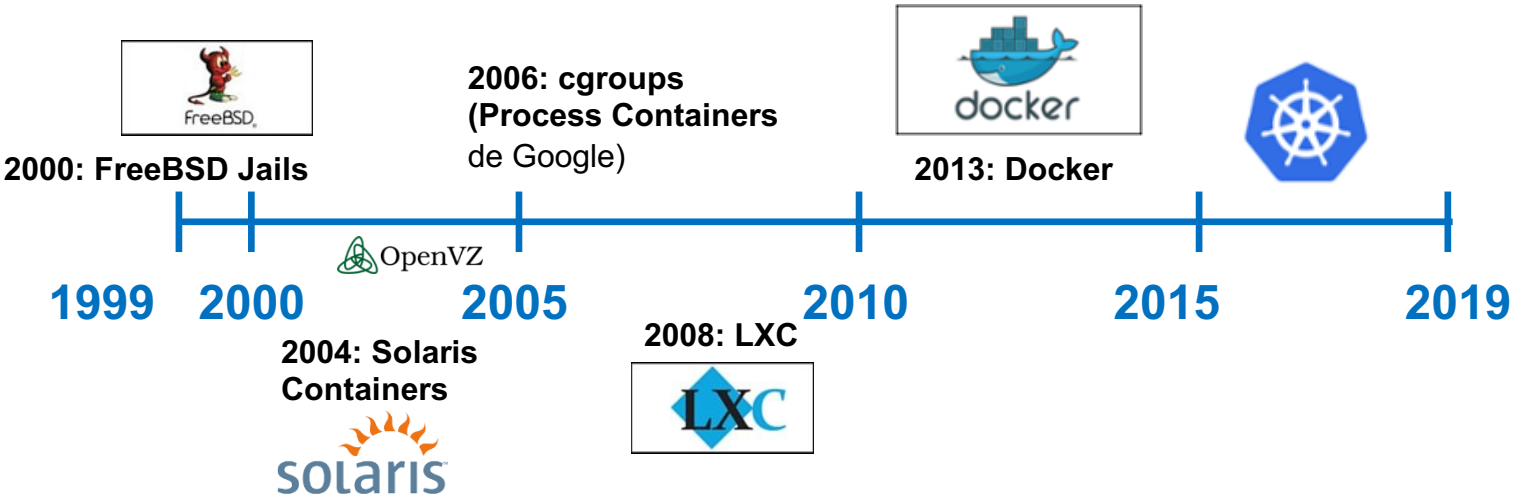
Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Otras soluciones (sin hipervisor)

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Otras soluciones: containers





Fuente:
A Brief History of Containers, Jeff Victor & Kir Kolyshkin, 2015

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

¿Qué es un container?

- Idea:
  - MV ligera (arranque rápido, clonado eficiente, mínima instalación ...)
  - Se busca una aproximación lo más próximo posible a los 3 principios de P&G
  - Estandarización, portabilidad

- Es virtualización a nivel de SO, pero no es una MV:
  - Usa el kenel del host
  - No puede arrancar un kernel propio (ni diferente)
  - ... ni tiene sus drivers
  - ... ni necesita algunos demonios o servicios (syslogd, cron, ...), ni un proceso PID 1 (init), ...

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

¿De qué está hecho un container?

- Basicamente: Es un conjunto de procesos visibles al *host* pero que mantienen cierto aislamiento entre ellos
- Container = *chroot on steroids*
- Conjuga 3 elementos (perspectiva Unix/linux):
  - Sandboxing:
    - chroot
  - Mapeo de identificadores:
    - namespaces
      - nsenter
  - Control de recursos
    - cgroups

Fuente: Container's Anatomy: Namespaces, cgroups,and some filesystem magic,Jérôme Petazzoni, 2015

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Container orchestration

- Gestión y control del ciclo de vida de los containers en especial en grandes infraestructuas.
  - Aprovisionamiento y despliege
  - Redundancia / diponibilidad
  - Equilibrado de carga, migración de containers entre hosts
  - Exposición de servicios hacia el exterior
  - Monitorización de recuros
  - Configuración

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Otras soluciones

- Distribución de linux bare metal
- Combina dos tecnologías de virtualización:
  - hipervisor KVM
  - containers LXC
- Gestor de MVs a través de un interfaz web

Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar