Introducción a la Virtualización v.20211006

Arquitecturas Virtuales

Depto. de Arquitectura de Computadores
Universidad de Málaga

Indice

- Conceptos básicos de virtualización
- Tipos de hipervisores
  - Hosted
  - Native o Bare Bones
- Implementación de los recursos virtuales
  - Virtualización de la arquitectura (CPU, RAM)
    - Soporte hardware a la virtualización
  - Virtualización del almacenamiento
  - Virtualización del interfaz de red (NIC)
  - *Snapshots* y Clones

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

Objetivos

- Definir los conceptos que vamos a usar durante todo el curso
- Conocer las funciones que ofrece una infraestructura de virtualización para empresas
- Comprender los mecanismos de implementación de dichas funciones
- Ser capaz de razonar sobre las implicaciones de la virtualización sobre el rendimiento observado
- Ser capaz de elegir la mejor configuración de un sistema para obtener un rendimiento adecuado en un escenario concreto

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

Conceptos básicos sobre virtualización

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

¿Qué es la virtualización?

- **Ofrecer recursos hardware ficticios** usando implementaciones software o hardware.
- El software **preexistente** escrito para el hardware real debe funcionar **sin modificación alguna** en el virtual.

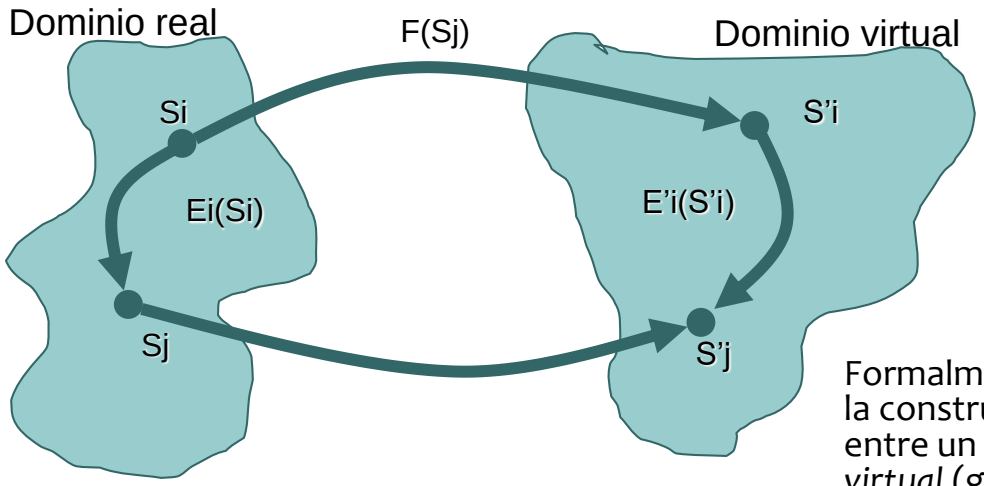


Ilustración: Advanced Operating Systems: Virtualization, Roy Campbell

Formalmente, la virtualización implica la construcción de un isomorfismo entre un sistema *real* (host) y otro *virtual* (guest)

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

¿Qué es la virtualización?

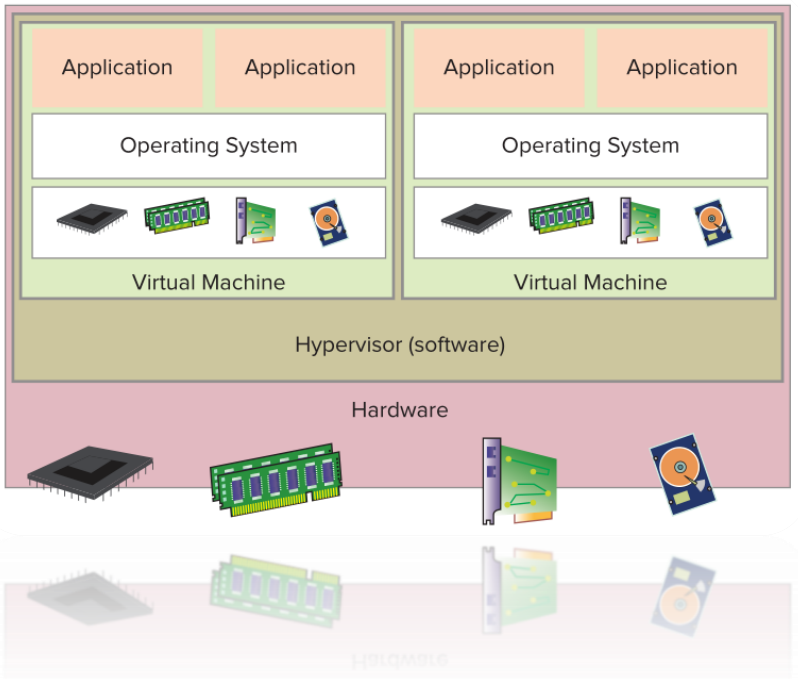
- **Ofrecer recursos hardware ficticios** usando implementaciones software o hardware.
- El software **preexistente** escrito para el hardware real debe funcionar **sin modificación alguna** en el virtual.
- La única diferencia que percibe el software son **temporizaciones atípicas** (debidas a la implementación y a la compartición de recursos con otras máquinas virtuales).
  - **Retardos adicionales** en la ejecución de operaciones de **E/S** y otro hardware.
  - **Pausas** en la ejecución de instrucciones en la **CPU** sin razón aparente.

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

Motivaciones

- Emulación de otras máquinas ó sistemas operativos
- Ejecución de software obsoleto (*legacy code*)
- Simulación y experimentación (prototipos, *sandboxes*, *what-if?*)
- Fuerte aislamiento (*isolation*) entre los *guests*:
  - separación de servicios en diferentes servidores virtuales

Ilustración: Virtualization Essentials, M. Portnoy

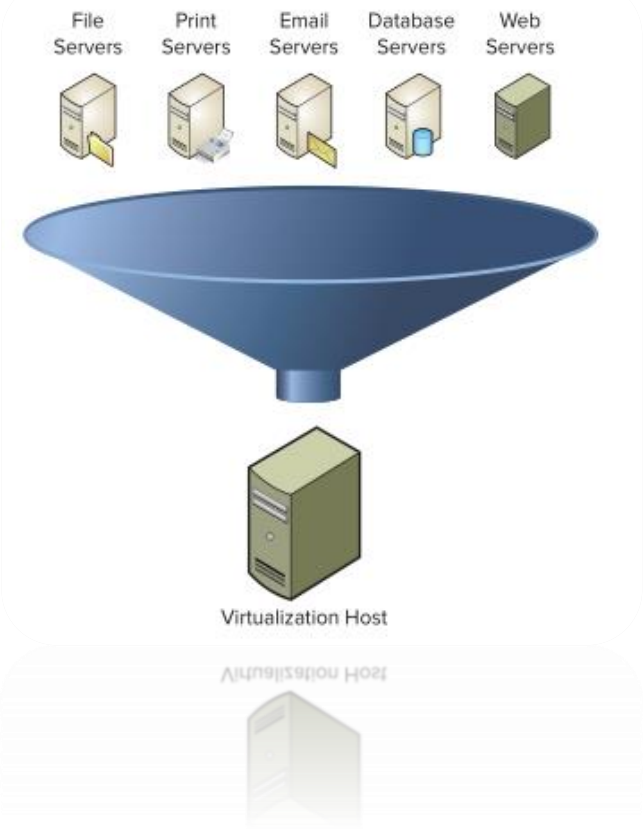


Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Más motivaciones

- Consolidación:
  - Las máquinas físicas son caras
  - Permite ejecutar varios SO sobre un mismo HW
  - Facilita el despliegue masivo (i.e. servidores web)
  - Infraestructuras a prueba de fallos (*failover*)
  - Permite la migración entre soportes físicos
  - Clonación de máquinas virtuales (respaldo)
  - Facilita el balanceo de carga
  - *Green computing*

Ilustración: Virtualization Essentials, M. Portnoy

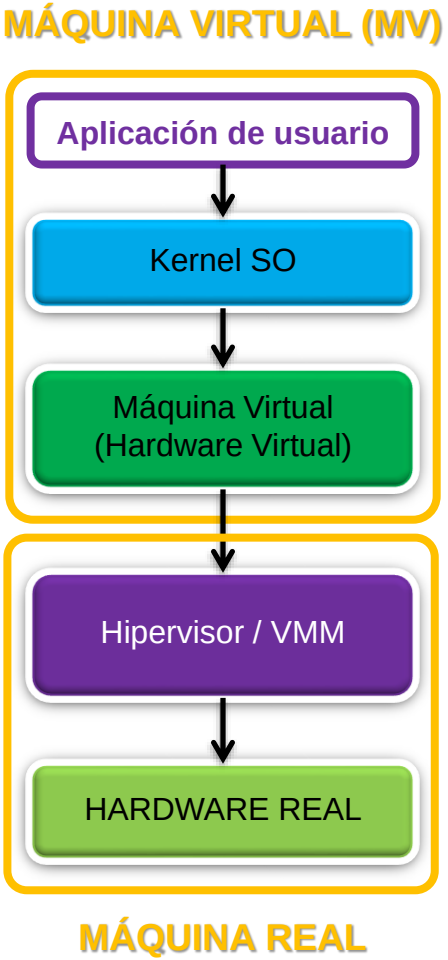


El balanceo de carga en máquinas virtuales es una técnica utilizada para distribuir el tráfico de red o la carga de trabajo entre varias máquinas virtuales

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

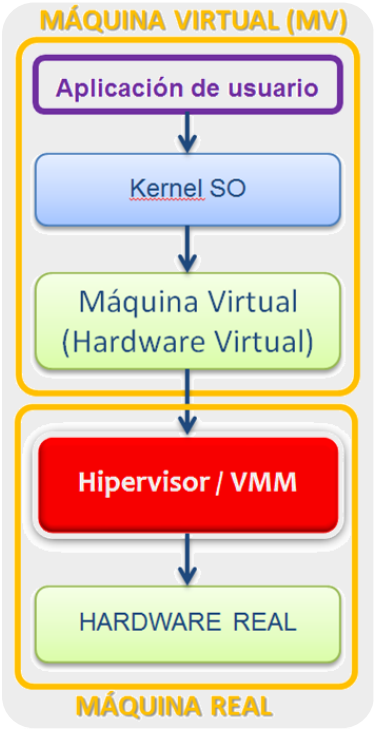
Máquinas virtuales: Terminología

- **Hipervisor** (ó **VMM**=Virtual Machine Monitor): Es el software de virtualización en sí (a veces llamado emulador) proporciona una abstracción de una máquina
- **Máquina virtual** (**VM**): que trabaja con recursos virtualizados: CPU, RAM, almacenamiento, red, etc.
- ***Guest***: El sistema operativo instalado en la máquina virtual
- ***Host:*** Un sistema operativo instalado en el hardware físico (asumiendo que contiene algún *guest*)



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

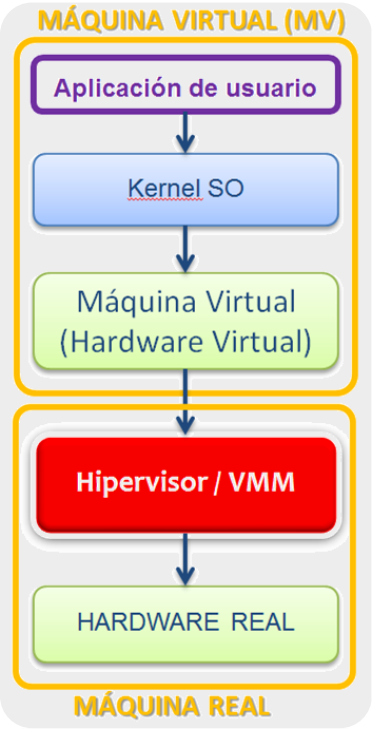
Hipervisor



- **Es todo el conjunto de software encargado de implementar la virtualización.**
- Los primeros sistemas operativos se denominaban “Supervisores”, de forma que el nombre de “Hipervisor” viene de su concepción como una evolución del Sistema Operativo.

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

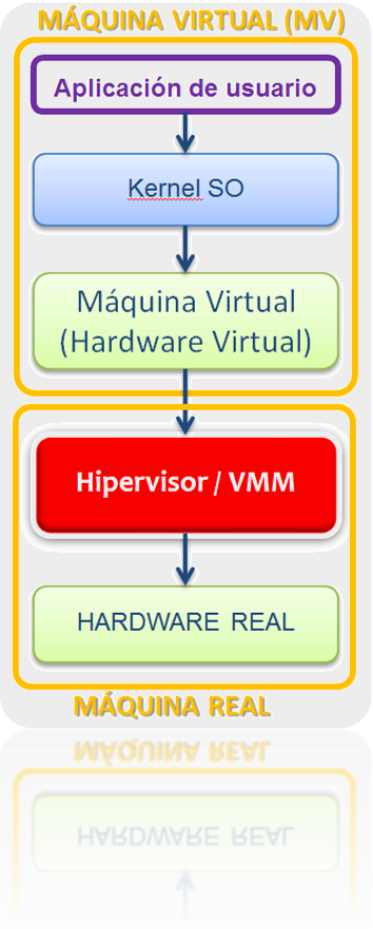
Hipervisor



- Es una capa software (aunque puede estar asistida por el hardware).
- El hipervisor tiene control completo de los recursos del sistema físico.
- Permite múltiples MVs ejecutarse sobre un mismo sistema físico.
- Debería permitir que las aplicaciones en las MVs se ejecuten sin modificación.
- Proporciona aislamiento (*isolation*): un programa que se ejecute en una MV no puede acceder a recursos que no se le hayan asignado.

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

Hipervisor



- Funciones:
  - *Dispatcher*: decide qué hacer tras un *trap* (interrupción)
  - *Allocator*: asigna recursos físicos (reales) a las MVs
  - *Intérprete*: simula las instrucciones que intercepta
- Su estructura y los aspectos que gestiona son muy similares a los del kernel de un S.O.:
  - Compartición de CPU entre máquinas virtuales
  - Compartición de memoria entre máquinas virtuales
  - Control de permisos y arbitraje del acceso a los periféricos, discos, red, …

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

Hipervisor vs Sistema Operativo

Hipervisor

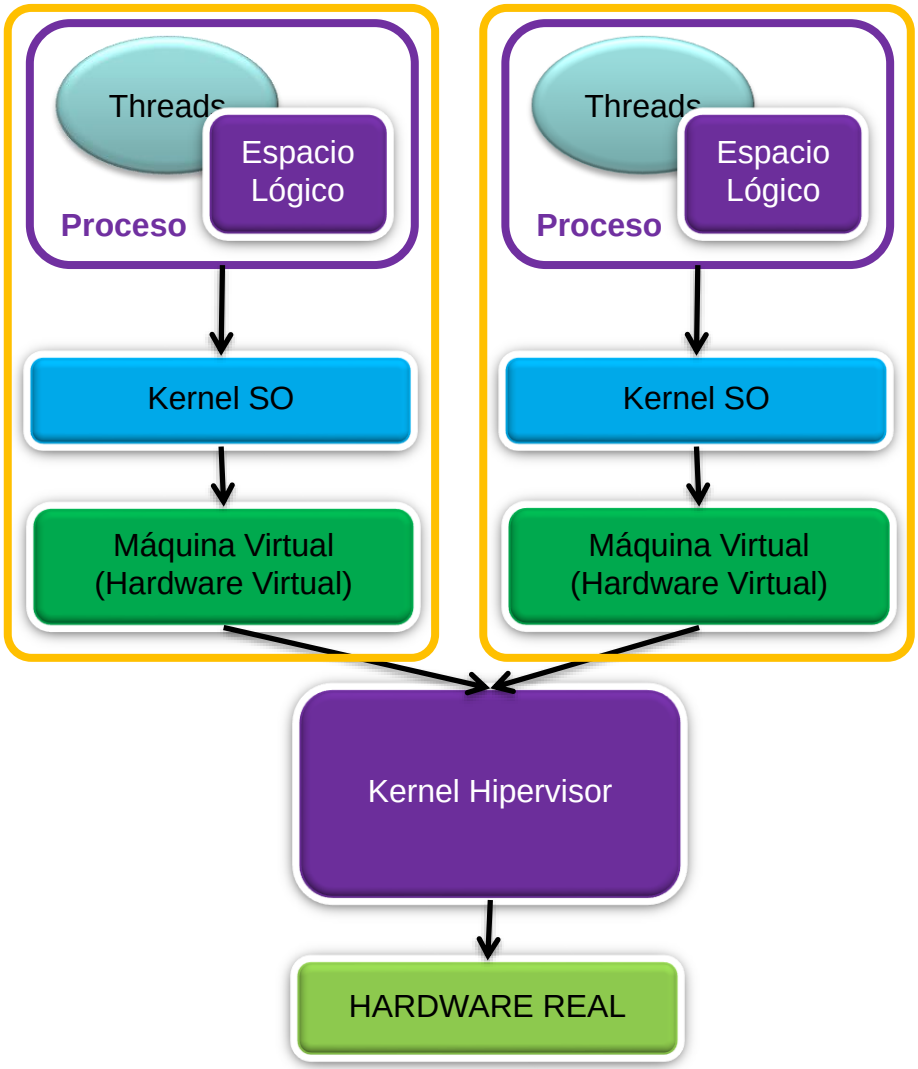
- Oculta el hardware real
- El hardware se gestiona en forma de recursos hardware virtuales:
  - CPU→CPU virtual
  - RAM→RAM virtual
  - Disco→Disco virtual
  - NIC→NIC virtual
- Los clientes del hipervisor son Máquinas virtuales
- El software que se ejecuta en las máquinas virtuales son Sistemas Operativos

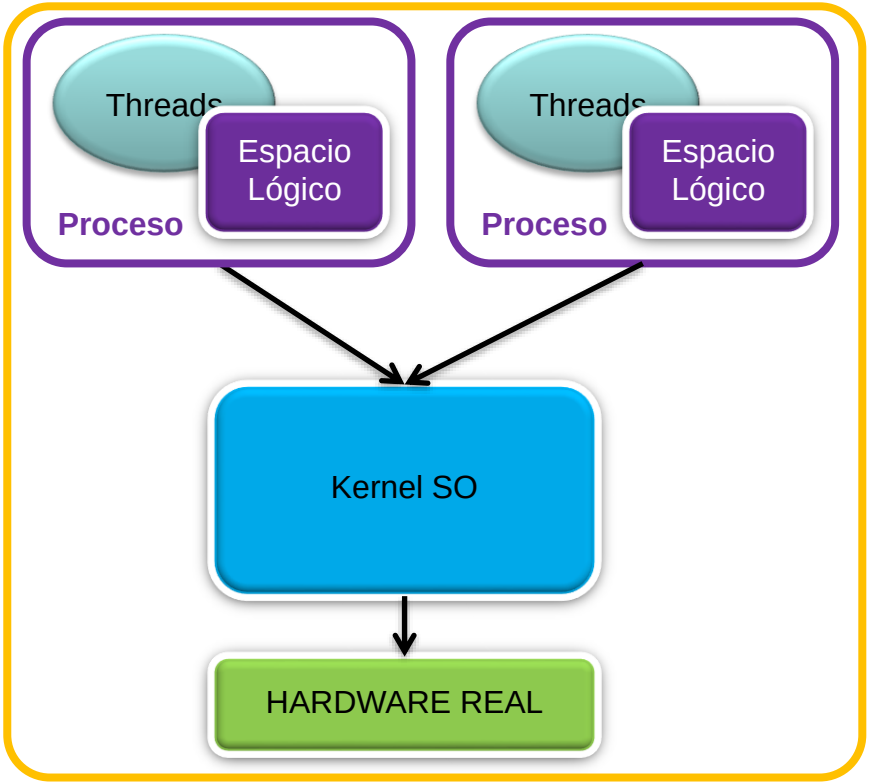
Sistema Operativo

- Oculta el hardware real
- El hardware se gestiona en forma de servicios de alto nivel:
  - CPU→Procesos
  - RAM→Espacio lógico
  - Disco→Ficheros
  - NIC→Sockets
- Los clientes del kernel son Procesos
- El software que se ejecuta en los procesos son Programas y Aplicaciones

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

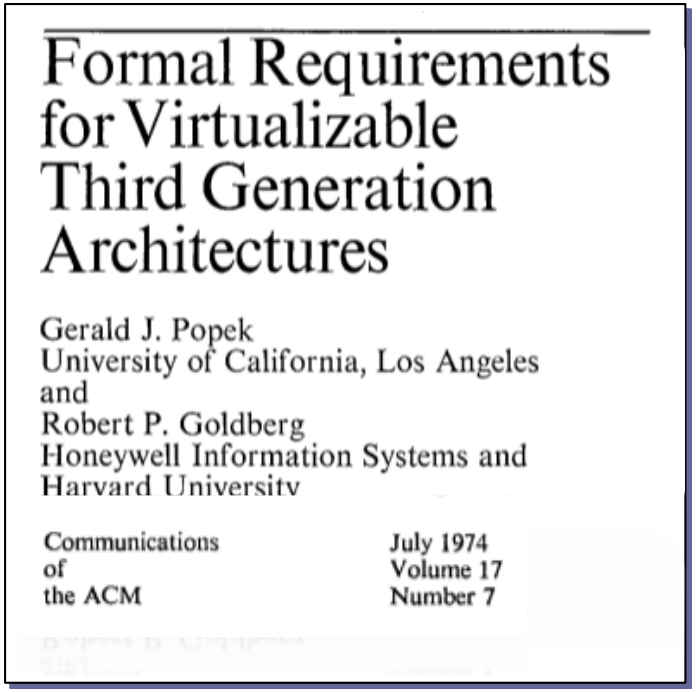
Hipervisor vs Sistema Operativo





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

Requerimientos formales



Formal Requirements for Virtualizable Third Generation Architectures

Gerald J. Popek
University of California, Los Angeles
and
Robert P. Goldberg
Honeywell Information Systems and
Harvard University

Communications of the ACM

July 1974
Volume 17
Number 7

- **1974** – Artículo de Popek & Goldberg
- Introduce los conceptos de máquina virtual (VM) e hipervisor (VMM):
  - Equivalencia: El programa en ejecución sobre una MV debe mostrar un comportamiento idéntico al de la máquina que se simula
    - Excepción: temporizaciones
  - Control de recursos: El hipervisor tiene el control completo de los recursos virtualizados
  - Eficiencia: una fracción importante de instrucciones debe ejecutarse sin la intervención del hipervisor

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

Popek & Goldberg: instrucciones

- Tipos de instrucciones en la máquina real:
  - Privilegiadas: generan un *trap* en modo usuario, y sólo se ejecutan correctamente en modo privilegiado
  - *Sensitive instructions*:
    - Control-sensitive: cambian la configuración de recursos del sistema
    - Behaviour-sensitive: su comportamiento depende de la configuración de recursos (e.g., si estamos en modo supervisor o usuario)
  - Instrucciones inocuas

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

Teorema 1: virtualización (estricta)

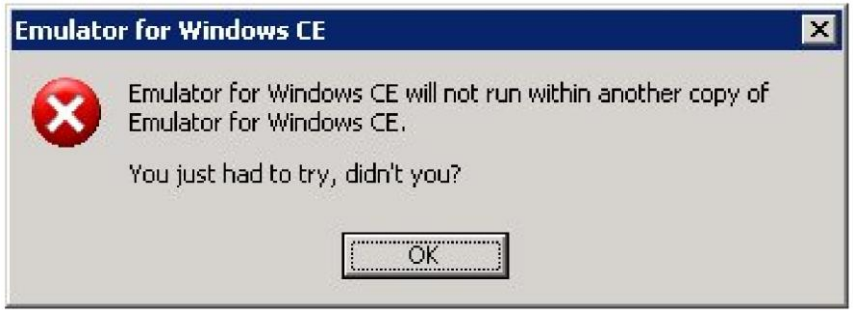
- "For any conventional third generation machine, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions"
  - De esta manera todas las instrucciones sensibles siempre generan un *trap* y ceden el control al hipervisor

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

Teorema 2: virtualización recursiva

- "A conventional third generation computer is recursively virtualizable if it is
  a) virtualizable and
  b) a VMM without timing dependencies can be constructed for it."

- La virtualización recursiva no es siempre posible



Ilustraciones extraídas de: Virtualizacion; Darren Alton.

Imagen: Cobb from Inception

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

Teorema 3: virtualización híbrida

- “A hybrid VMM may be constructed for any third generation machine in which the set of user sensitive instructions are a subset of the set of privileged instructions”
  - El teorema 1 es una condición suficiente de virtualizabilidad
  - Siempre nos quedará la *emulación*
  - El teorema 3 relaja la condición de *eficiencia*

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

Taxonomía de los hipervisores

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

Tipos de máquinas virtuales

- ***Kind I: Full Machine Virtualization***
  - ***Hipervisores tipo 1 (Native / Bare Metal)*** El VMM se ejecuta directamente sobre la máquina física. Los SO invitados están bajo el control del VMM.
  - ***Hipervisores tipo 2 (Hosted)***: El VMM es una aplicación ordinaria de un SO host, pero emula una máquina virtual sobre la que puede ejecutar un SO invitado
- ***Kind II: Paravirtualization***
  - El VMM se ejecuta directamente sobre el hardware (aunque no lo emula), como un SO, y ofrece un interfaz con la máquina real de manera que el SO invitado debe ser adaptado a dicho interfaz
- ***Kind III:***
  - *Language Runtimes*, ej. Java Virtual Machine
  - *Containers* (no realmente una máquina virtual, sino virtualización *a nivel de sistema operativo*)

Fuente: http://srl.cs.jhu.edu/courses/600.418/lectures/VM.pdf

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

Tipos de virtualización

- ***Full virtualization***
  - El SO está en una capa de abstracción independiente, completamente desacoplado del HW subyacente a través del hipervisor
  - El SO no es consciente de que está siendo virtualizado
  - Por tanto no requiere modificación

- ***Para virtualization***
  - Existe comunicación explícita entre el SO guest y el hipervisor, con el objetivo de ser más eficaz
  - Implica modificar el kernel del SO guest, reemplazando las instrucciones no-virtualizables por hipercalls que comunican directamente con el hipervisor
  - El hipervisor también proporcionan hipercalls para otras operaciones críticas del kernel (manejo de memoria, manejo de las interrupciones, información de reloj, …)

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

Tipos de hipervisores

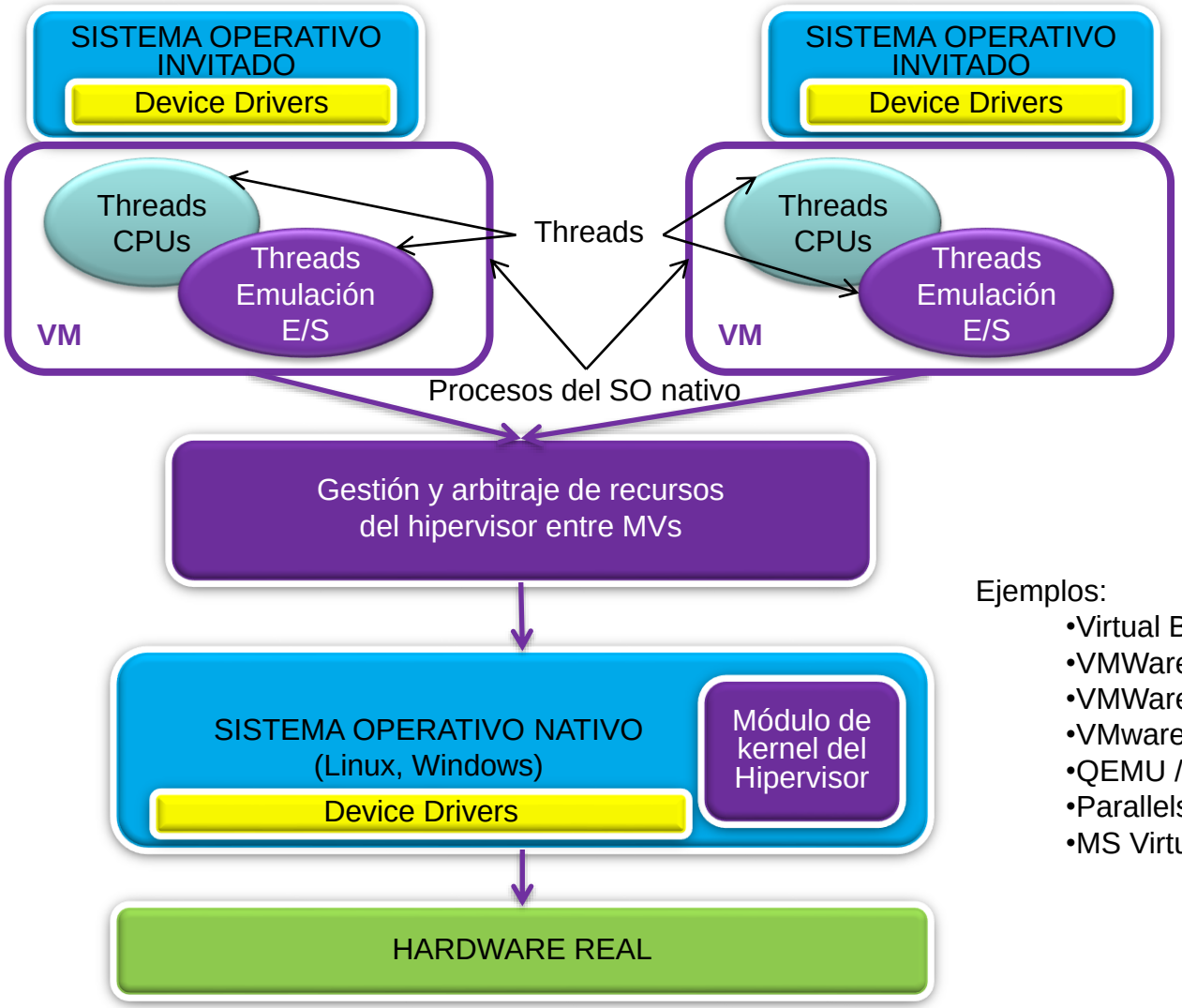
- ***Hosted***
  - Ejecuta el código de la CPU virtual dentro de un proceso normal de un S.O. en modo usuario.
  - Las instrucciones privilegiadas son interceptadas por un módulo insertado en el núcleo del S.O.
  - Simula las operaciones de E/S usando servicios del S.O. real (p.e. ficheros).
  - Ejemplo: VMWare Workstation, Virtual Box, QEMU

- ***Native*** o ***Bare Metal***
  - El hypervisor es un kernel propietario sin S.O. debajo
  - Simula las operaciones de E/S directamente sobre sus propios drivers que controlan el hardware.

- ***Native*** **+ máquina virtual privilegiada** para ejecutar los drivers
  - Intercepta las operaciones de E/S y las implementa a través de una máquina virtual que tiene acceso al hardware real.
  - Se pueden aprovechar los drivers de un S.O. existente instalado en la máquina virtual privilegiada
  - Ejemplo: Hyper-V, VMWare ESX, Xen (Ambos usan un sistema Linux para aprovechar sus drivers)

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

Hosted Hypervisor

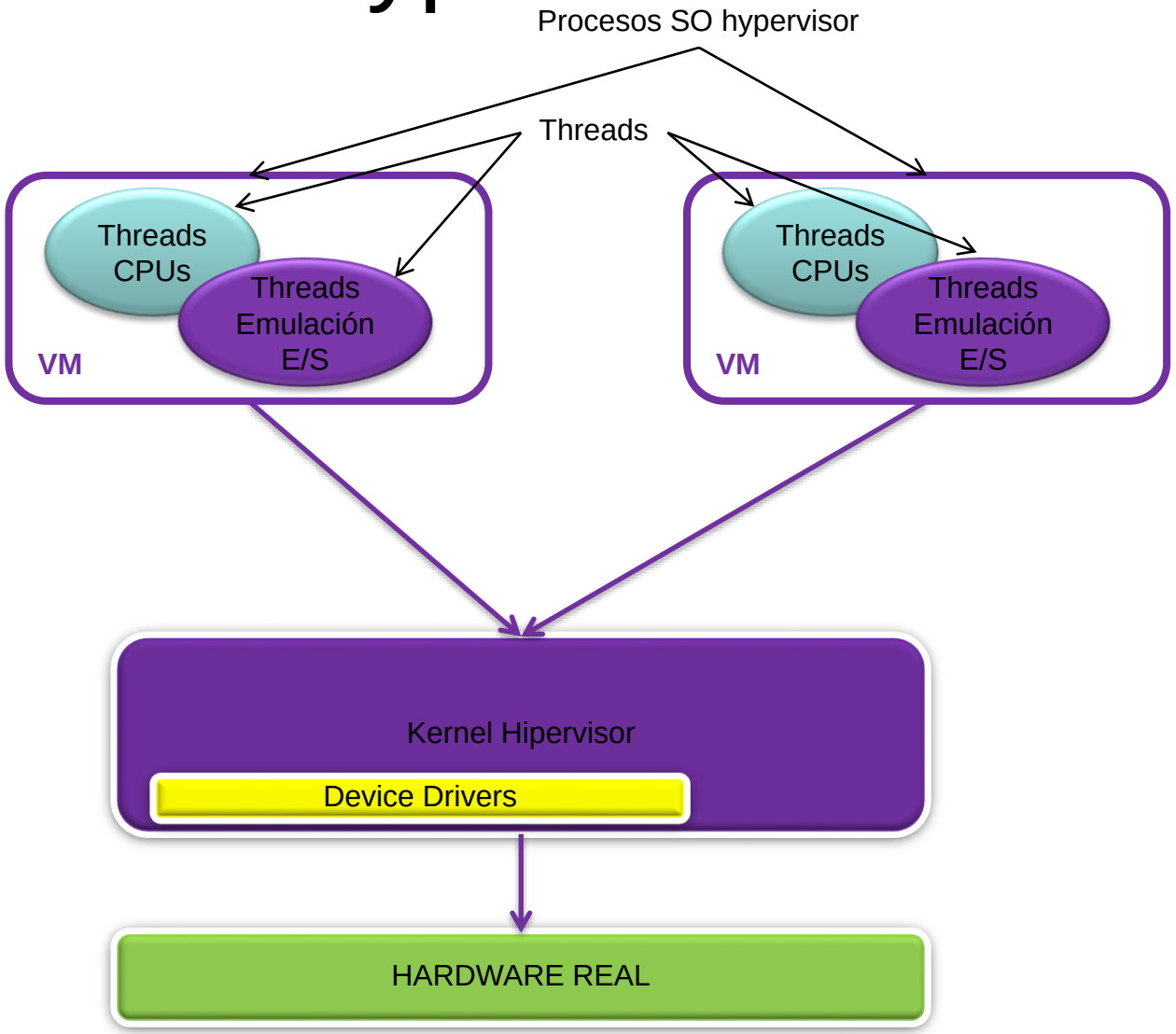


Ejemplos:

- Virtual Box
- VMWare Worskstation
- VMWare Player
- VMware Fusion
- QEMU / KVM
- Parallels Desktop for Mac
- MS Virtual PC (retirado)

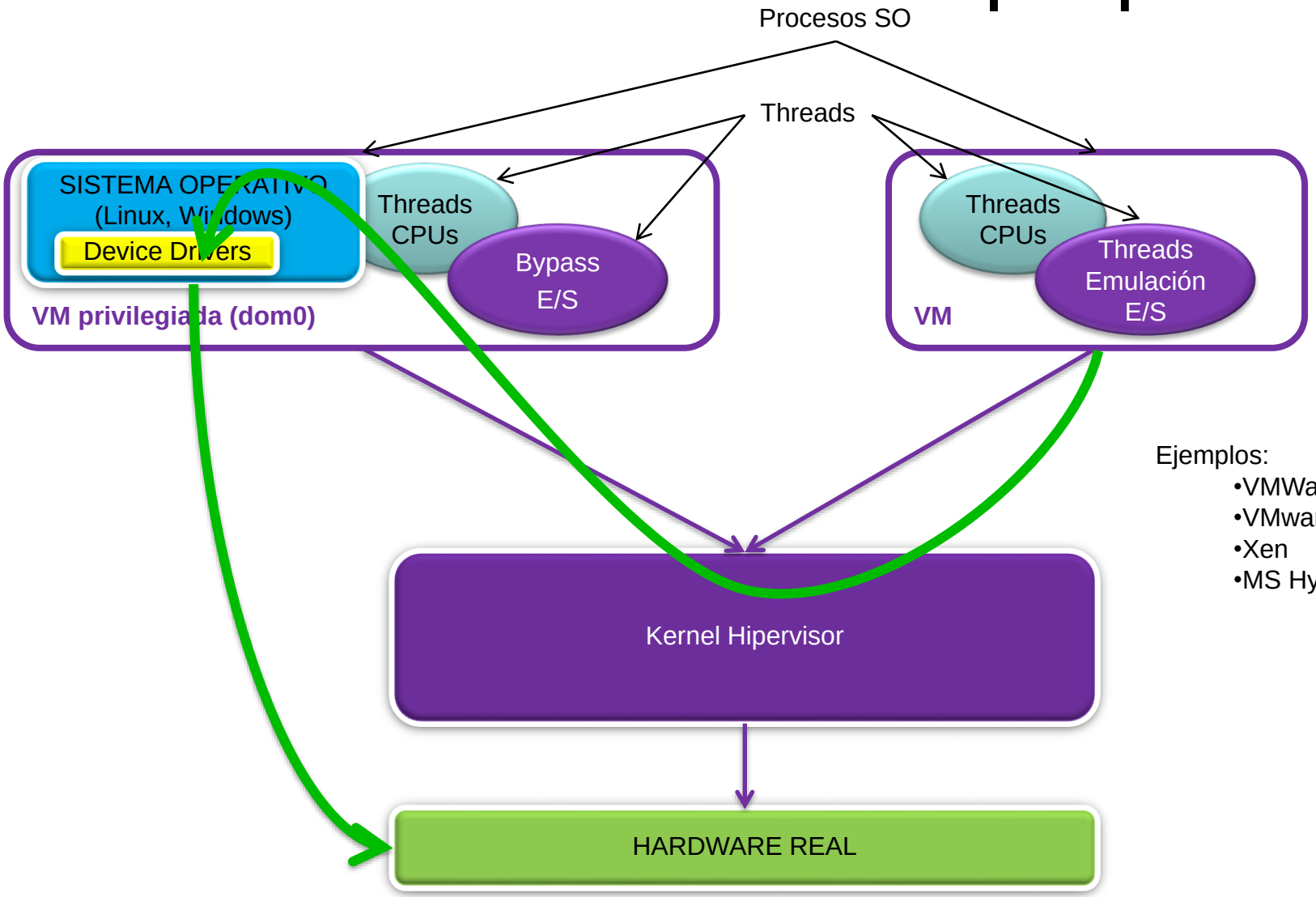
Universidad de Málaga
Guillermo Pérez Trabado, Eladio Gutiérrez Carrasco, Julián Ramos Cózar

Native Hypervisor



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

Native sin drivers propios



Ejemplos:

- VMWare ESXi
- VMware vSphere
- Xen
- MS Hyper-V

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