

INTRO A LA VIRTUALIZACIÓN

ALBERTO MOLINA COBALLES

IES GONZALO NAZARENO

5 DE OCTUBRE DE 2020



(cc) 2011 Alberto Molina Coballes

Esta presentación se distribuye bajo licencia
Creative Commons Reconocimiento 3.0 España.

[http:
//creativecommons.org/licenses/by-sa/3.0/es/](http://creativecommons.org/licenses/by-sa/3.0/es/)

Este documento incluye algunas partes de:
[El arte de virtualizar](#), de Miguel Vidal y José Castro.



- 1 Introducción
 - Conceptos previos
- 2 Principales técnicas de virtualización
 - Emulación
 - Virtualización completa
 - Virtualización por hardware
 - Paravirtualización
 - Virtualización ligera
 - Otros tipos de virtualización
- 3 Cuadro comparativo



INTRODUCCIÓN



- Objetivo** Aumentar el rendimiento del hardware disponible incrementando el tiempo de procesamiento de un equipo, ya que habitualmente se desaprovecha gran parte.
- Método** Instalar varios sistemas operativos en una misma máquina real para que funcionen como máquinas virtuales.



¿PARA QUÉ SE UTILIZA?

- Consolidación de servidores
- Aislamiento e independencia de servicios y contenidos
- Laboratorio de pruebas
- Mantenimiento de sistemas antiguos
- Virtualización de arquitecturas de las que no se dispone
- Sistemas distribuidos
- Herramienta de aprendizaje
- Cloud computing



Principales ventajas

- Importante ahorro económico
- Seguridad
- Mayor aprovechamiento de recursos
- Migración en vivo
- Importante ahorro energético

Principales inconvenientes

- Muchos sistemas dependen de un sólo equipo físico
- Penalizaciones en rendimiento



- Al sistema operativo que ejecuta el software de virtualización se le conoce como anfitrión (host).
 - ▶ El anfitrión controla el hardware real.
- Al sistema operativo virtualizado se le conoce como invitado o huésped (guest).
 - ▶ Puede haber varios huéspedes en un mismo anfitrión.
 - ▶ Los huéspedes no deben interferir entre ellos ni con el anfitrión.



- Al software de virtualización se le llama:
 - ▶ Hipervisor.
 - ▶ Virtual Machine Manager (VMM).
- El VMM o Hipervisor corre como parte del sistema operativo del anfitrión (o es el anfitrión)
- A una instancia del hardware virtualizado se la conoce como Máquina Virtual o VM.
- Los sistemas operativos huéspedes corren dentro de una VM.



- Los hipervisores permiten que diferentes sistemas operativos, tareas y configuraciones de software coexistan en una misma máquina física.
- Abstraen los recursos físicos de la máquina anfitriona para las distintas máquinas virtuales.
- Garantizan un nivel de aislamiento entre los invitados.
- Proporcionan una interfaz única para el hardware.



Hay dos clases de hipervisores:

Tipo 1, nativo o bare-metal el hipervisor es una capa entre el hardware y el sistema operativo.

- Al sistema operativo se le llama Dominio de Control, Dominio Principal o Domo y corre sobre el hipervisor.
- Los huéspedes son Dominios Lógicos.

Tipo 2 o hosted el hipervisor es una capa de software que corre sobre el sistema operativo anfitrión.



- Desde 2005, Intel y AMD han añadido soporte hardware para la virtualización.
- Intel Virtualization Technology (VT) codename Vanderpool
- AMD Virtualization (AMD-V) codename Pacifica
- Añaden una funcionalidad específica para permitir a los hipervisores un rendimiento mayor en virtualización completa.
- La virtualización completa es más sencilla de implementar.

¿Tiene estas extensiones mi CPU?

```
egrep --color '(vmx|svm)' /proc/cpuinfo
```



PRINCIPALES TÉCNICAS DE VIRTUALIZACIÓN



PRINCIPALES TÉCNICAS DE VIRTUALIZACIÓN

Emulación La máquina virtual simula un hardware completo y el SO huésped sin modificar se ejecuta dentro de la VM.

Completa El sistema operativo anfitrión simula el hardware (utilizando un hipervisor tipo II) y sobre él se ejecutan los sistemas operativos huésped sin modificar

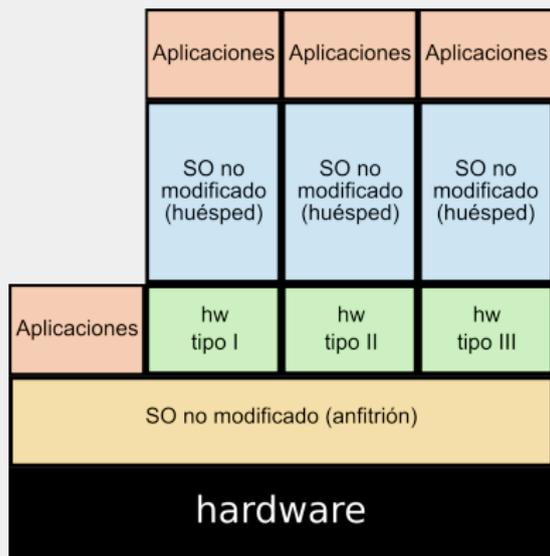
Por hardware o acelerada Extensión de la virtualización completa, que es más eficiente al utilizar hardware (CPU) adaptado.

Paravirtualización Utiliza un hipervisor tipo I sobre el que se ejecutan todos los dominios.

Contenedores o ligera El SO está modificado para permitir múltiples procesos en diferentes espacios de usuario aislados unos de otros, cada uno con su configuración de red.



EMULACIÓN



Hardware Convencional, además pueden emularse otras arquitecturas.

Ejemplos Qemu

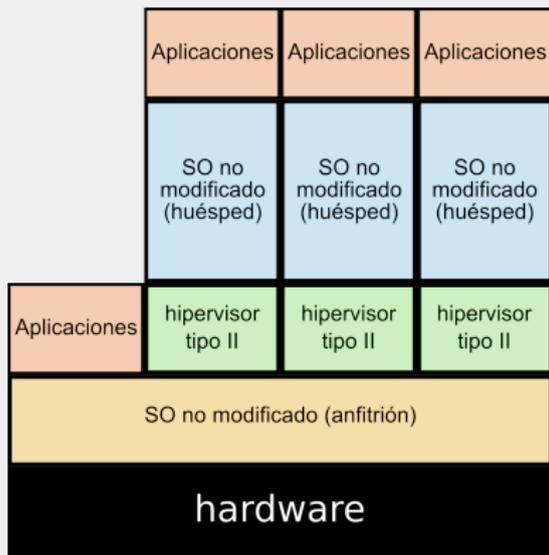
Ventaja Facilidad de uso

Defecto Bajo rendimiento

Utilización Ejecutar sistemas sobre otras arquitecturas (e.g. ARM sobre x86)



VIRTUALIZACIÓN COMPLETA



Hardware Convencional

Hipervisor tipo II

Ejemplos VMWare Server, VirtualBox, Parallels Desktop, Virtual PC

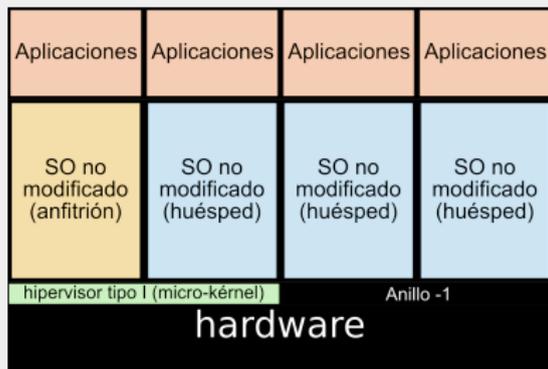
Ventaja Facilidad de uso

Defecto Bajo rendimiento

Utilización Virtualización en equipos convencionales



VIRTUALIZACIÓN POR HARDWARE



Hardware Extensiones en CPU (Intel-VT, AMD-V)

Hipervisor tipo I

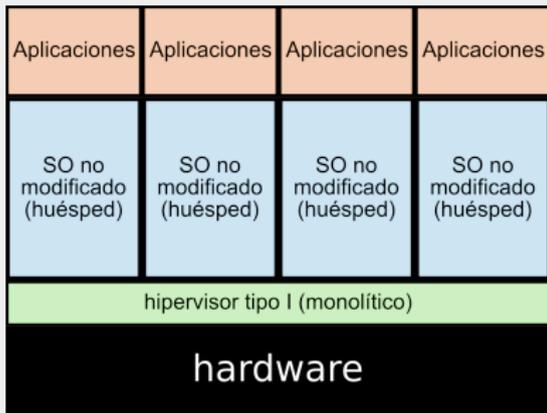
Ejemplos KVM, Xen HVM, Hyper-V

Ventajas Alto rendimiento

Defecto No sirve hw convencional

Utilización Servidores/CPD





Hardware Específico

Hipervisor tipo I

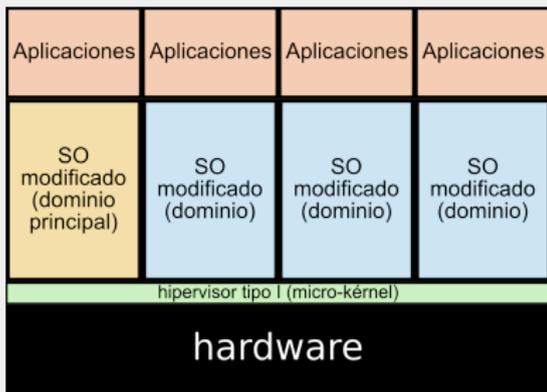
Ejemplos VMware ESX(i)

Ventajas SO huésped no modificado, alto rendimiento

Defecto Poco hardware soportado

Utilización Servidores/CPD





Hardware Depende
Hipervisor tipo I

Ejemplos Xen, Hyper-V

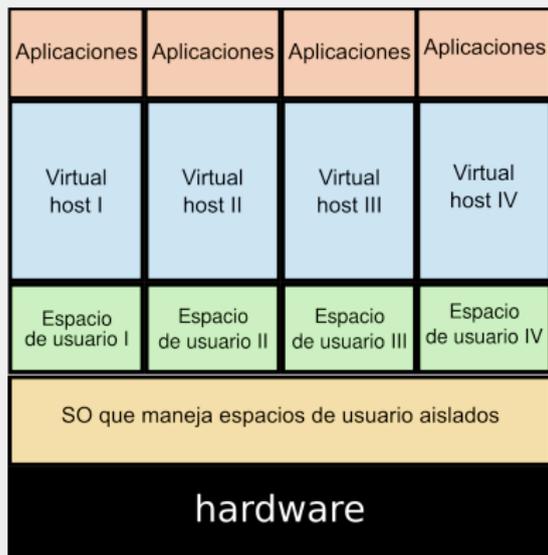
Ventajas Alto rendimiento

Defecto SO modificado

Utilización Servidores/CPD



VIRTUALIZACIÓN LIGERA O DE SISTEMA OPERATIVO



Hardware Convencional

Ejemplos Jails, Containers, Virtuozzo, LXC, ...

Ventajas Alto rendimiento, fácil implementación

Defecto Aislamiento entre los virtual host, todos los SO iguales.

Utilización Servidores/CPD



- Virtualización de bibliotecas: biblioteca Wine (subconjunto de la API de Win32 para poder ejecutar aplicaciones Windows)
- Virtualización de aplicación: entorno de ejecución virtual (con una API para la ejecución en diferentes plataformas). Ejemplo: Java Virtual Machine.
- Virtualización de escritorio: se implementa el escritorio como servicio. Ejemplo: SunVDI.



CUADRO COMPARATIVO



CUADRO COMPARATIVO

Nombre	Virtualización					Licencia
	Emu	Comp	Para	Hw	Cont	
Qemu	✓	✗	✗	✗	✗	Libre
Xen	✗	✗	✓	✓	✗	Libre
VirtualBox	✗	✓	✗	✗	✗	Mixta
LXC	✗	✗	✗	✗	✓	Libre
Jails	✗	✗	✗	✗	✓	Libre
Containers	✗	✗	✗	✗	✓	Libre
KVM	✗	✗	✗	✓	✗	Libre
VMWare ESX(i)	✗	✗	✓	✗	✗	Privativa
Hyper-V	✗	✗	✓	✗	✗	Privativa
XenServer	✗	✗	✓	✓	✗	Libre?
Virtuozzo	✗	✗	✗	✗	✓	Privativa

Fuente: http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines

