

# Sistemas Operativos II

---

## *Módulo V*

# VIRTUALIZACIÓN

1

## Temas del Módulo V

- **Introducción.**
- **Virtualización.**
  - Requisitos para la virtualización.
  - Tipos de virtualización.
  - Hipervisores. Contenedores.

2

## Introducción

- En ciertas ocasiones, una empresa tiene una multicomputadora sin darse cuenta.
- Si cuentan con un servidor de correo, un servidor web, un servidor de base de datos, y un servidor FTP, etc., conectados a una red de alta velocidad, en definitiva, se tiene una MC.
- Por lo general, estos servidores ejecutan en máquinas separadas, ya que una sola no puede manejar la carga.
- Pero otra razón importante es la confiabilidad: si una máquina falla, se “cortará” ese servicio, pero los demás no se verán afectados.
- Si bien de esta manera se logra una tolerancia a fallos, la solución con múltiples computadoras para cada servicio es costosa y difícil de administrar, por la cantidad de equipos involucrados.

3

## Introducción

- Una solución alternativa es la tecnología de máquinas virtuales (en adelante **vm**, del inglés virtual machine), también conocida como **virtualización**.
- Esta tecnología permite que una sola computadora contenga varias vm, cada una ejecutando un SO distinto.
- La tolerancia a fallas se mantiene: una falla en una vm no afecta el funcionamiento de las demás, pero a un costo menor y con una administración más fácil.
- Claro está que, si ahora falla la computadora que contiene todas las vm, el resultado es catastrófico.
- Sin embargo, la razón por la que se usa la virtualización es porque la mayoría de las fallas no se deben a hardware defectuoso, sino al software poco confiable y con errores, en especial, los SO.

4

## Introducción

- Con la virtualización, el único software que se ejecuta en kernel es el hipervisor, el cual tiene 100 veces menos líneas de código que un SO completo, y por ende tiene 100 veces menos errores.
- Otra ventaja es que al tener menos máquinas físicas hay un ahorro en hardware y electricidad, y se ocupa menos espacio en la oficina.
- Por otro lado, también es muy útil para la ejecución de aplicaciones heredadas en SO (o versiones de SO) que ya no tienen soporte en hw actual.
- El desarrollo de software es otro uso importante de la virtualización. Si un programador quiere asegurar la compatibilidad de su app con varios SO, puede crear vm's, cada una con el SO que necesite.
- Un SO ejecutando en una vm es llamado **invitado** (guest, en inglés), mientras que el SO que ejecuta en una máquina física es llamado **anfitrión** (host, en inglés).

5

## Virtualización

### Definición de virtualización.

- Pero ¿qué es la virtualización?
- Una definición sencilla es la siguiente:
  - Creación a través de software de una versión virtual de algún recurso tecnológico. (Turban, E; King, D; Lee, J; Viehland, D; 2008).
- Los recursos tecnológicos a los que se refiere esta definición son:
  - Hardware.
  - Sistema Operativo.
  - Almacenamiento.
  - Red.

6

## Virtualización

### Requisitos para la virtualización.

- Por un lado, es necesario contar con una capa de abstracción: un **software que implementa la virtualización** denominado **VMM** (Virtual Machine Monitor, monitor de máquinas virtuales ). El VMM también es llamado **hipervisor** (hypervisor, en inglés). Es el encargado de crear la “ilusión” de que cada vm ejecuta directamente sobre el hardware real.
- Por otro lado, se necesita un **CPU virtualizable**. Esto implica que el set de instrucciones debe tener la siguiente organización:
  - **Instrucciones sensibles:** acceden directamente al subsistema de E/S y a la MMU. Sólo pueden ejecutarse en modo kernel.
  - **Instrucciones privilegiadas:** son aquellas que, ejecutadas en modo usuario, deberían ocasionar una trap (por ejemplo, las que cambian valores en los registros del CPU). **Deben incluir al conjunto de instrucciones sensibles.**
  - **Instrucciones en modo usuario:** todo el resto del set de instrucciones.

7

## Virtualización

### Tipos de virtualización.

- Se pueden definir dos tipos básicos de virtualización:
  - **Virtualización de plataforma:** consiste en la creación de las vm utilizando un hipervisor, simulando algún tipo de hardware. Hay varios tipos de virtualización de plataformas:
    - Emulación.
    - Virtualización completa.
    - Paravirtualización.
    - Virtualización asistida por hardware.
    - Virtualización a nivel de SO.
  - **Virtualización de recursos:** consiste en simular recursos tales como:
    - **Almacenamiento:** discos o arreglos de discos (RAID, NAS, SAN).
    - **Redes:** utilizando tecnologías como VLAN, SDN, NFV, etc.

8

## Virtualización

### Tipos de virtualización. (cont.)

#### ▪ Emulación.

- La máquina virtual simula un **hardware completo**, admitiendo SO para arquitecturas de CPU completamente diferentes al CPU real.
- Esto implica que es posible ejecutar programas en plataformas de hw diferentes para los que fueron escritos.
- La ventaja de los emuladores es que permiten simular hardware que no está disponible físicamente. La desventaja es que el rendimiento es bajo, porque el costo de computación es muy alto.
- Algunos ejemplos de emuladores:
  - QEMU.
  - MAME.
  - Bochs.

9

## Virtualización

### Tipos de virtualización. (cont.)

#### ▪ Virtualización completa.

- La vm simula un hardware suficiente para poder permitir a un SO invitado sin modificar, correr de forma aislada sobre el mismo tipo de CPU que la máquina física.
- Entonces, **tanto el SO anfitrión como el invitado, ejecutan sobre el mismo CPU**. El hipervisor atrapa las instrucciones sensibles del SO invitado, y emula su ejecución (**traducción binaria**).
- La ventaja es que se logra un rendimiento superior a la emulación, ya que no es necesario simular el hardware completo.
- La desventaja es que no se pueden emular otras arquitecturas.
- Productos como VirtualBox, MS Hyper-V, VMware Workstation, etc., utilizan este tipo de virtualización.

10

## Virtualización

### **Tipos de virtualización.** (cont.)

#### ▪ **Paravirtualización.**

- Se eliminan (algunas de) las instrucciones sensibles, lo que significa que [se modifica el código del SO invitado](#).
- Entonces, en lugar de instrucciones sensibles, se realizan llamadas al hipervisor. Estas llamadas forman una API: **VMI** (Virtual Machine Interface, interfaz de máquina virtual).
- Con esto se tiene como ventaja un mejor rendimiento respecto de la virtualización completa, ya que no es necesaria la traducción binaria.
- Por otro lado, otro lado, se puede implementar con un CPU que no tenga soporte para virtualización.
- La desventaja es que hay que modificar el código del SO, lo cual es muy complicado, sobre todo en SO que no son de código abierto.

11

## Virtualización

### **Tipos de virtualización.** (cont.)

#### ▪ **Virtualización asistida por hardware.**

- No tiene traducción binaria ni VMI: [el procesador da soporte a las instrucciones sensibles](#) del SO invitado, asistiendo al hipervisor. Por otro lado, no es necesario modificar el código del SO.
- Este soporte de hw fue introducido por Intel (2005) con su tecnología VT-x (Vanderpool), y con AMD (2006) con AMD-V (Pacifica).
- Con esto se logra un rendimiento muy superior a la virtualización completa, e incluso facilita enormemente su implementación.
- Este es el tipo de virtualización más utilizado para la consolidación de servidores, por todas las ventajas mencionadas.
- VMware ESXi, Proxmox VE, Citrix Hypervisor, son los productos más populares que utilizan esta tecnología.

12

## Virtualización

### Tipos de virtualización. (cont.)

- **Virtualización a nivel de sistema operativo.**
  - También llamada **virtualización ligera**. El SO anfitrión virtualiza el hw de manera que crea vm's que ejecutan aisladas pero sobre el mismo hardware. De hecho, el SO invitado utiliza el mismo kernel del SO anfitrión.
  - El kernel permite múltiples instancias aisladas de espacios de usuario. Se utilizan más que nada para aislar aplicaciones. Puede no haber un hipervisor.
  - A las vm creadas con esta técnica se les llama **contenedores** (docker, en inglés). Tienen la ventaja de tener un rendimiento muy alto, cercano al nativo, por compartir el kernel.
  - La desventaja es que no permiten virtualizar SO diferentes.

13

## Virtualización

### Hipervisores.

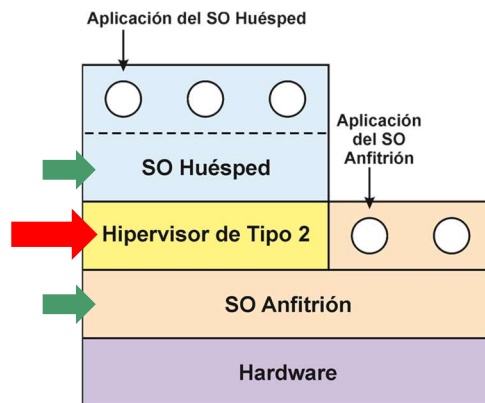
- Existen dos tipos: Tipo 2 (*hosted*) y Tipo 1 (*bare metal*).
- **Hipervisor Tipo 2 (hosted).**
  - Ejecuta en modo usuario, como una aplicación cualquiera, por encima de un SO anfitrión.
  - Tanto el SO invitado como sus aplicaciones, también ejecutan modo usuario.
  - Típicamente utilizan virtualización completa, aunque en algunos productos, también utilizan emulación.

14

## Virtualización

### Hipervisores. (cont.)

#### ▪ Hipervisor Tipo 2 (hosted). (cont.)



15

## Virtualización

### Hipervisores. (cont.)

#### ▪ Hipervisor Tipo 1 (bare metal).

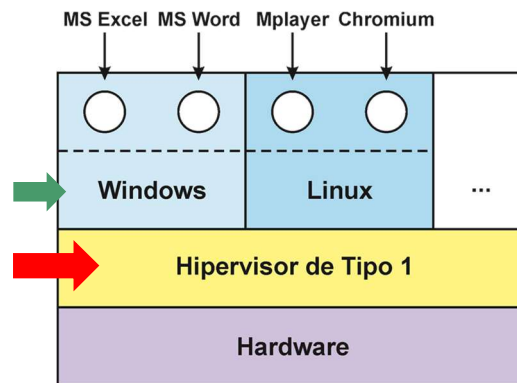
- También llamado **nativo**, ejecuta en modo kernel, directamente por encima del hardware. Esto es, no hay ningún SO entre el hipervisor y el hardware.
- Es el único software que ejecuta en modo kernel. Todos los SO invitados y sus aplicaciones ejecutan en modo usuario.
- Inicialmente, estos hipervisores utilizaban virtualización completa y paravirtualización. Actualmente, utilizan virtualización asistida por hardware, y en algunos productos, además dan soporte para contenedores.
- En 2007, el kernel de Linux incorporó el módulo KVM (Kernel-based Virtual Machine), siendo una alternativa gratuita de virtualización.

16

## Virtualización

### Hipervisores. (cont.)

- **Hipervisor Tipo 1 (bare metal).** (cont.)



17

## Virtualización

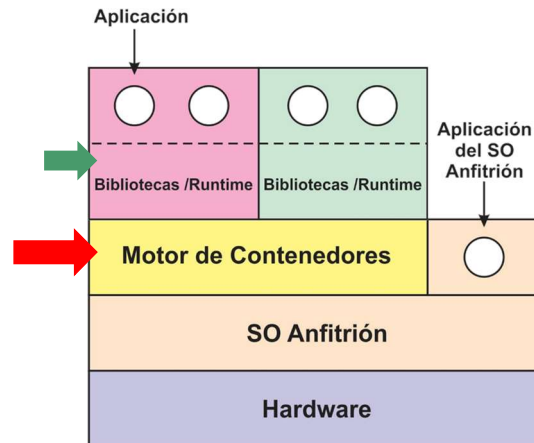
### Contenedores.

- Mencionamos anteriormente que las vm creadas con virtualización a nivel de SO se llamaban **contenedores**.
- Estos se utilizan para **aislar aplicaciones**.
- Comparten el kernel del SO anfitrión. Los hipervisores que utilizan el módulo KVM, también dan soporte para crear contenedores, como es el caso de Proxmox VE.
- Son más “livianos” que las vm:
  - Menor carga a la máquina física.
  - Menos almacenamiento requerido
  - Menor tiempo inicio.
  - No necesitan definir de antemano los recursos necesarios.

18

## Virtualización

### Contenedores. (cont.)



19

***Fin del Módulo V***

20