Configuración y Optimización de Sistemas de Cómputo
Virtualización

Master Universitario en Ingeniería Informática
Depto. de Informática de Sistemas y Computadores (DISCA)
Universidad Politécnica de Valencia

Virtualización

- Modo protegido
- Virtualización
  - Tipos de Virtualización
  - Paravirtualización
  - Contenedores vs Maquinas Virtuales
- Soporte Hardware para Virtualización
  - MMU, IOMMU
  - Aceleradores
  - Interrupciones

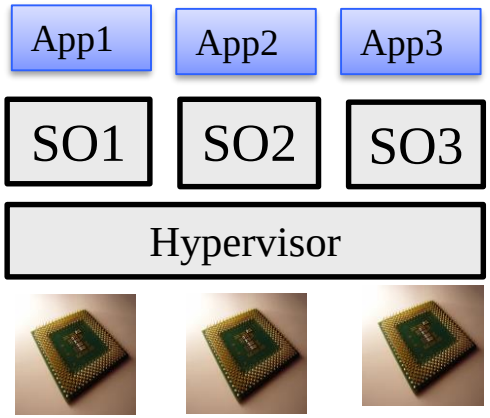
Virtualización

- El objetivo es poder compartir de manera segura y eficiente los recursos de un computador
- Los primeros virtualizadores utilizaban virtualización por software
  - Lenta
  - Tiene limitaciones a la hora de virtualizar todos los recursos
- Procesadores modernos incluyen soporte en el ISA para facilitar la virtualización

Virtualización

- En un sistema virtualizado nuestras aplicaciones no se comunican directamente con el hardware
- Una capa SW hace de intermediario para nuestra aplicación

Sistema Físico



Sistema Virtualizado

Virtualización

- Bare-metal (modo maquina)
  - Modo de ejecución de software sobre la maquina física
  - El software puede acceder a todo el hardware sin restricciones
  - Modo de ejecución “peligroso”
    - No hay protección de memoria
    - Cualquier dispositivo puede modificar datos en el sistema

Maquinas Virtuales

- Una Maquina Virtual ofrece al usuario un conjunto de recursos hardware virtualizados :
  - Procesador/es
  - Memoria
  - Periféricos (dispositivos de entrada/salida)
- Un monitor de la maquina virtual (VMM o también hypervisor) es la pieza software que proporciona la abstracción de una maquina virtual
- **Gerald J. Popek** and **Robert P. Goldberg** definen de forma formal en 1974 los requisitos que tiene que cumplir un sistema para ser virtualizado
  - "Formal Requirements for Virtualizable Third Generation Architectures

Maquinas Virtuales

- Propiedades que tiene que cumplir una maquina virtual
  - Equivalencia / Fidelidad
    - Un programa que se ejecuta sobre un hypervisor debe exhibir un comportamiento esencialmente idéntico al de la maquina física
  - Control de Recursos/ Seguridad
    - El hypervisor tiene que tener complete control sobre los recursos virtualizados
  - Eficiéncia / Rendimiento
    - Un porcentaje significativo de las instrucciones maquina deben ejecutarse sin la intervención del hypervisor

Maquinas Virtuales

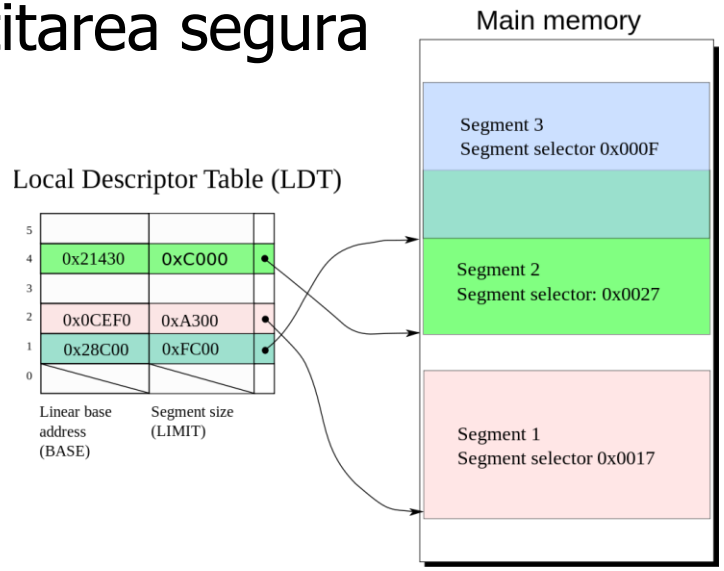
- Conjuntos de instrucciones relevantes
  - Instrucciones Privilegiadas
    - Aquellas que resultan en un **trap** si se ejecutan en modo usuario y no hacen **trap** cuando están en modo supervisor
  - Instrucciones Sensibles de Control
    - Aquellas que tartan de modificar la configuración de los recursos del sistema
  - Behavior sensitive instructions
    - Aquellas cuyo comportamiento o resultado depende de la configuración/estado de los recursos (p.ej de los contenidos del relocation register of del modo de ejecución del procesador)

Maquinas Virtuales

- **Teorema 1**. Un hypervisor efectivo puede construirse si el conjunto de instrucciones sensibles de la CPU es un subconjunto del conjunto de instrucciones privilegiadas
  - Para crear un hypervisor es suficiente que todas las instrucciones que pueden afectar al correcto funcionamiento del hypervisor (instrucciones sensibles) siempre realicen un trap y pasen el control al hypervisor. "Trap and Emulate" virtualization.
  - Las instrucciones no privilegiadas se ejecutan de forma nativa
- **Teorema 2**. Un computador es recursivamente virtualizable si:
  - Es virtualizable y un hypervisor sin dependencias temporales puede ser construido para ello
  - Algunas arquitecturas como x86 (antes de soporte hardware de virtualización) no cumplen esta condición. En este caso técnicas diferentes como "binary translation" remplazando aquellas instrucciones sensibles que no generan traps
- **Teorema 3**. Una maquina virtual hibrida puede ser construida para un computador en el cual es conjuto de instrucciones de usuario sensibles son un subconjunto del conjunto de privilegiadas

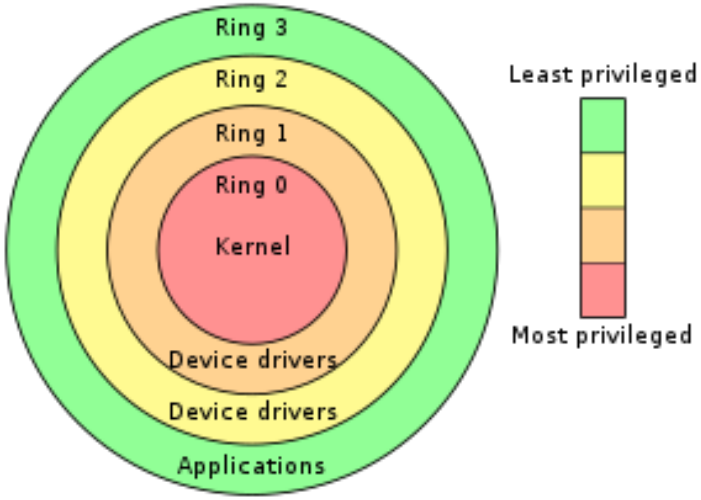
Modo Protegido (x86)

- Surge como mecanismo de protección para permitir la ejecución segura de diferentes tareas
  - Proporciona el soporte de memoria virtual
  - Soporte de ejecución multitarea segura
  - Paginación
- En modo "real" accedemos a memoria física



Anillos de Protección (x86)

- Clasificación de los niveles de privilegio de ejecución
  - 0 el más privilegiado
    - Kernel mode en Linux
  - 3 el menos privilegiado
    - Espacio de usuario



https://en.wikipedia.org/wiki/Protection_ring

Virtualización Software (x86)

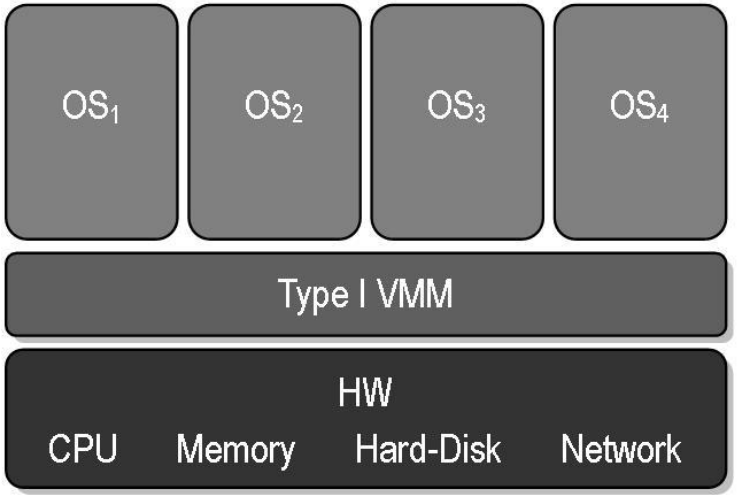
- La virtualización del modo protegido (x86) necesita las siguientes técnicas
  - *Binary translation* para modificar el comportamiento de ciertas instrucciones sensibles (que ahora se ejecutan en un modo de privilegio menor).
  - Creación de "Shadowed Structures" en software para ciertos componentes como por ejemplo la MMU (memory management unit). Creación de "shadowed page tables" para evitar que el SO pueda accede a la MMU directamente (sin control por parte del hypervisor)
  - Emulación de dispositivos de E/S: Dispositivos no soportados en el SO huesped (guest) tienen que ser emulados por software el SO anfitrión (host).

Virtualización Hardware

- Introducción de instrucciones y soporte especifico para mejorar la eficiencia de la virtualización
  - Diferentes modos de ejecución
  - Instrucciones para cambio de modos y replicación de estructuras hardware
- x86
  - Mejorar el problema de las instrucciones privilegiadas
    - EL SO huesped perceive su ejecución con todos los privilegios mientras que el SO hospedador sigue protegido
  - Soporte hardware virtualización de la MMU
    - Eliminación de las SW shadowed structures
  - Intel VT-x, AMD-V
    - El hypervisor se ejecuta en un nivel -1 → SO huésped en nivel 0

Hypervisor Tipo I

- Un hipervisor de tipo 1 se ejecuta directamente en el hardware físico
- Interacciona con la CPU, memoria y disco duro
- Ejemplos: Xen, KVM, Vware ESXi, Hyper-V, Jauilhouse

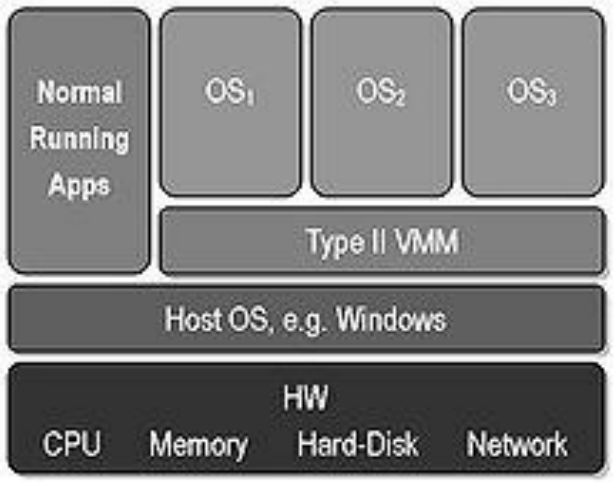


Hypervisor Tipo I

- Ejecución independiente
  - Recursos prácticamente disjuntos
- Mejor rendimiento
  - No hay sobrecarga de un SO adicional
- Mejores garantías de rendimiento
  - La asignación de recursos permite que las interferencias entre diferentes maquina virtuales se minimice
- Difícil compartir recursos entre maquinas virtuales

Hypervisor tipo II

- Se ejecuta sobre un SO como una aplicación más
- No interactúa directamente con el SW
- Ejemplos: VMWare Workstation, VirtualBox, QEMU



Hypervisor tipo II

- Más fáciles de manejar/instalar
  - Su instalación se puede hacer sobre un SO existente
- Peor estabilidad y rendimiento
  - El aislamiento de recursos es menor
  - Mayor variabilidad de rendimiento
- Fácil compartición de datos entre maquinas virtuales
- Insuficientes para aportar una virtualización segura y fiable

Mecanismos de Virtualización

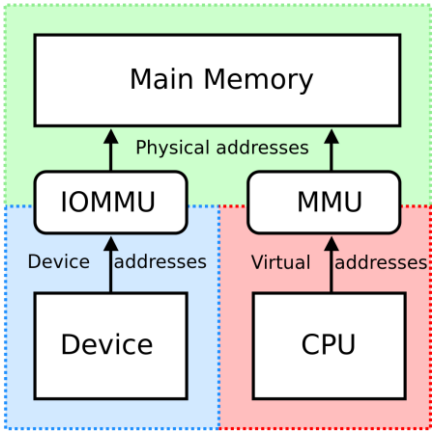
- Virtualización Completa
  - El hipervisor simula un hardware suficiente para permitir un sistema operativo no adaptado que es ejecutado de forma aislada.
    - Ejemplos: VirtualBox, HyperV, VMWare
- Virtualización Parcial (Paravirtualización)
  - El hipervisor ofrece una interfaz especial para acceder a los recursos. El sistema operativo de la máquina virtual tiene que ser adaptado usando llamadas especiales (hypercalls).
    - Ejemplos: Xen, L4
- Emulación
  - El hipervisor imita o suplanta vía software una arquitectura al completo
    - Ejemplos: QEMU, NAME, Wine

Soporte HW virtualización

- La CPU implementa ciertas instrucciones que facilitan la virtualización y extiende unidades HW
  - Intercepción de instrucciones
  - Virtualización MMU (memory management unit)
  - Virtualización Interrupciones
  - Inter-partition communications
- ¿Que pasa con otros recursos compartidos de un nodo ?
  - Periféricos, aceleradores, …

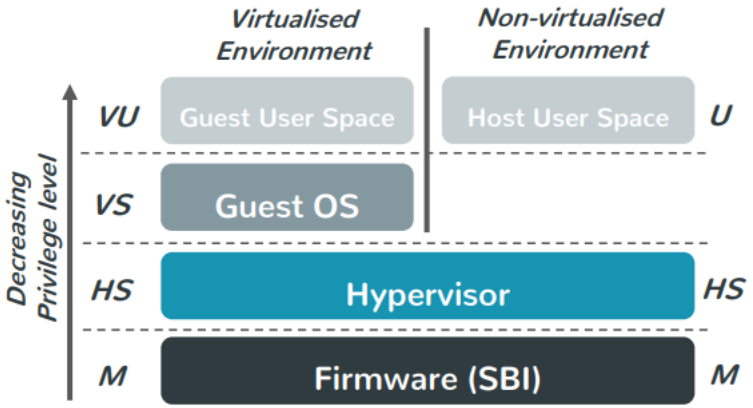
Soporte HW virtualización

- IOMMU
  - Es un dispositivo que protege la memoria frente a accesos por parte de dispositivos externos
  - Permite la virtualización eficiente de dispositivos
  - Introduce un sobrecoste en el rendimiento



RISC-V Execution Modes H-extension

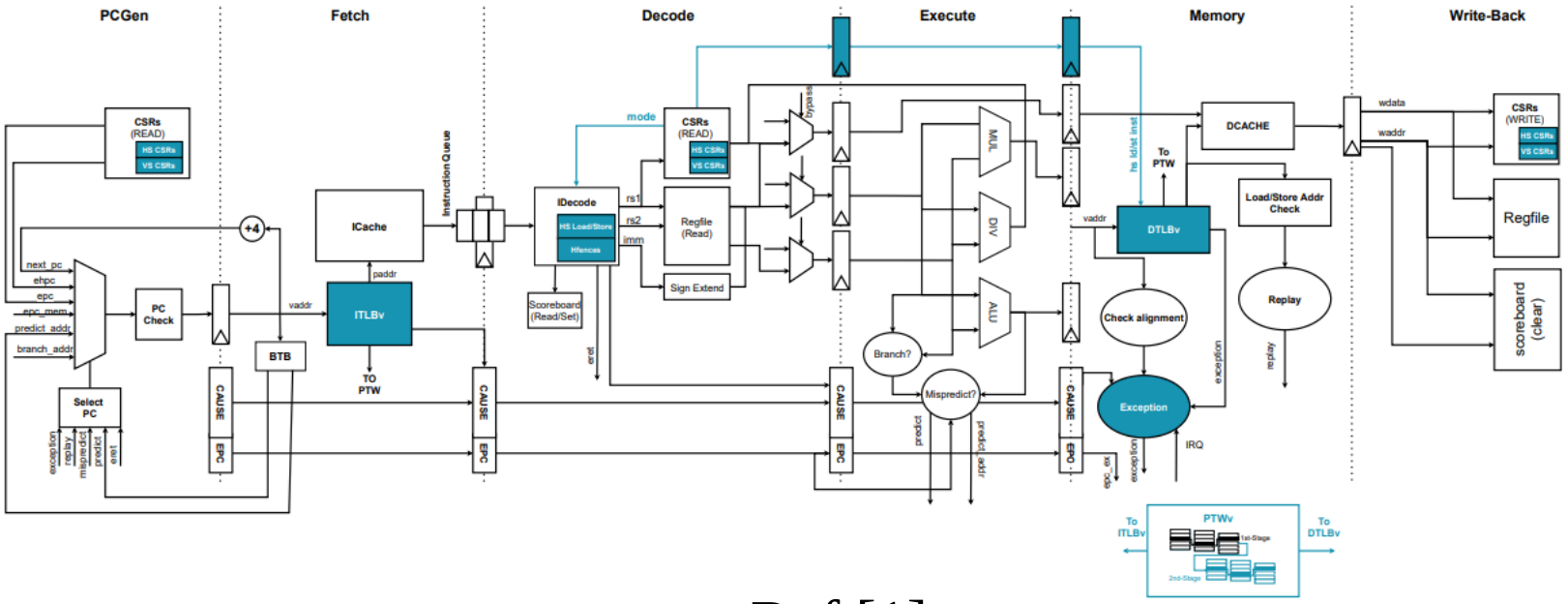
- La arquitectura RISC-V proporciona soporte hardware para virtualización
- El conjunto de instrucciones y cambios necesarios en el conjunto de instrucciones privilegiadas se define en la extensión (H)
  - Nuevos modos de ejecución
    - HS (hypervior)
    - VS (Virtual super user)
    - VU (virtual user)



Ref [1]

RISC-V Virtualization Support

- Pipeline Modifications



Ref [1]

RISC-V Virtualization support

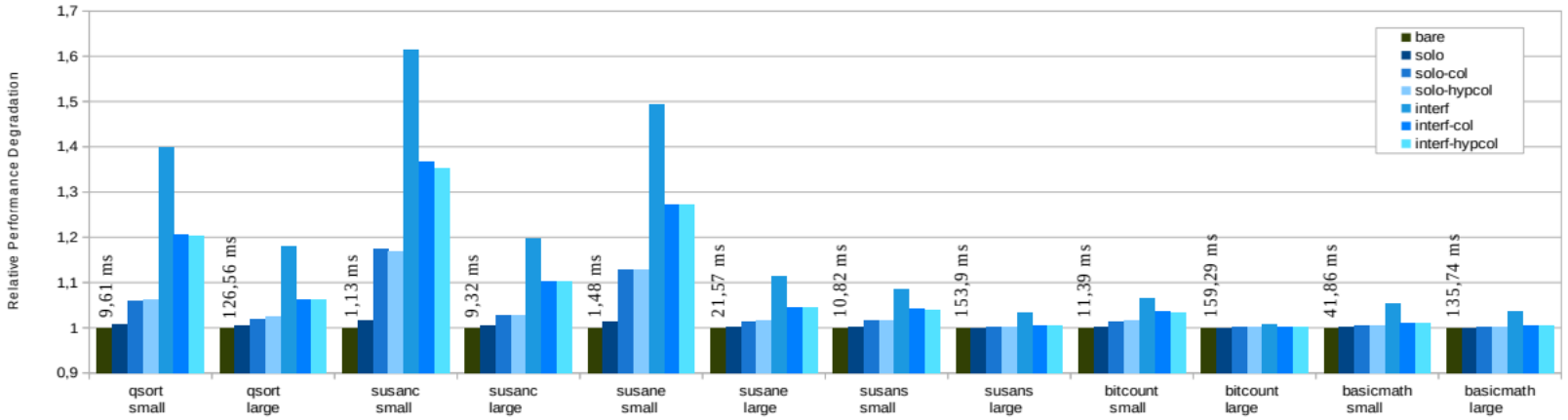
- Hardware Overheads
  - Cores and Interrupt Controlers

		Dual-Core	Quad-Core	Six-Core
Rocket Cores	LUTs	50922/**11%**	101744/**12%**	152957/**12%**
	Regs	25086/**30%**	50172/**30%**	75258/**30%**
CLINT	LUTs	68/**375%**	196/**296%**	269/**373%**
	Regs	194/**297%**	324/**336%**	454/**277%**
PLIC	LUTs	90/**140%**	144/**236%**	220/**263%**
	Regs	83/**325%**	116/**412%**	149/**460%**
Others	LUTs	11207/**2%**	13242/**3%**	91821/**0,5%**
	Regs	4257/**0,1%**	4628/**0,2%**	4728/**2%**
Total	LUTs	62287/**11%**	115356/**11%**	167753/**11%**
	Regs	29620/**27%**	55250/**28%**	80589/**29%**

Ref [1]

RISC-V Virtualization support

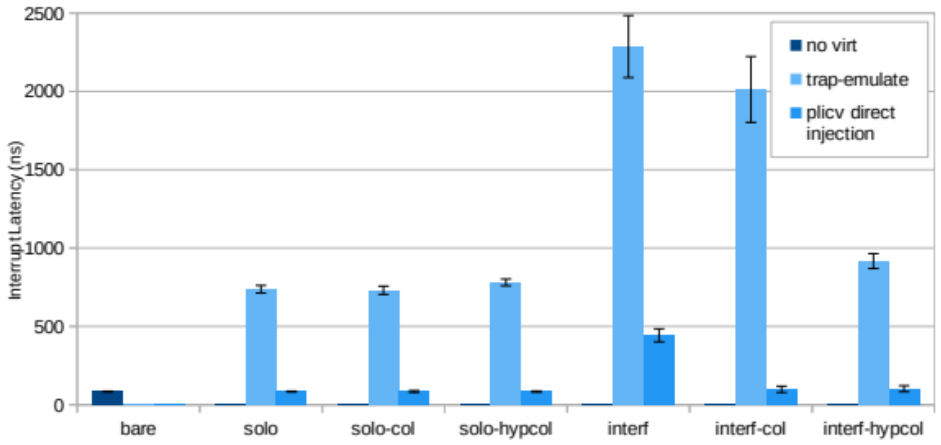
- Rendimiento en cargas de trabajo “normales” es cercano al rendimiento nativo



Ref [1]

RISC-V Virtualization support

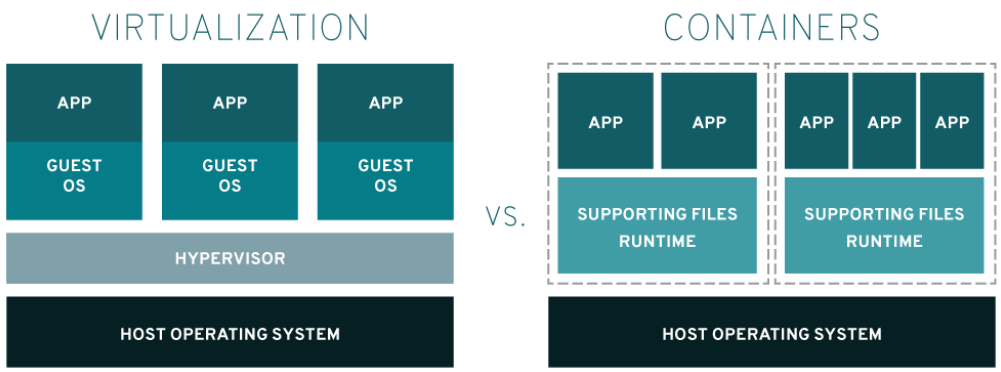
- Rendimiento de las interrupciones con soporte hardware de virtualización es cercano al nativo



Ref [1]

Maquinas Virtuales vs Contenedores

- Los contenedores contienen un microservicio o una aplicación y todo lo que necesita para ejecutarse
- Los contenedores se basan en el uso de imágenes
  - archivo basado en un código que incluye todas las bibliotecas y las dependencias que se despliega sobre una maquina o conjunto
- Maquinas virtuales (Hardware Virtualization)
  - Aislamiento de recursos
- Containers (OS level Virtualization)
  - Aislamiento de procesos



https://www.redhat.com/es/topics/containers/containers-vs-vms

Maquinas Virtuales vs Contenedores

- ¿Cuándo utilizar una tecnología u otra?
  - Contenedores
    - Copias multiples de un aplicación especifica
    - Aplicaciones que no necesitan rendimientos estrictos
  - Maquinas Virtuales
    - Despliegue de multiples aplicaciones o sistemas con necesidades de diferentes SOs
    - Cuando la seguridad es una prioridad (mayor aislamiento)

https://www.redhat.com/es/topics/containers/containers-vs-vms

Vulnerabilidades

- La virtualización (tipo I) en arquitecturas con soporte de virtualización proporcionan seguridad a nuestras aplicaciones
  - Particionan recursos
  - Imposibilitan la comunicación entre maquinas virtuales
- Limitaciones
  - Bugs: Pueden existir bugs que pueden ser explotados para romper la virtualización
  - Canales de ataque lateral
    - Muchos recursos siguen siendo compartidos
      - Caches, entradas TLBs, Interconexiones
  - Ataques de DoS
  - https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=hyper-V

Bibliografia

- [1] Bruno Sá, José Martins and Sandro Pinto. "A First Look at RISC-V Virtualization from an Embedded Systems Perspective". In IEEE Transactions on Computers, 2022
- [2] Popek, G. J.; Goldberg, R. P. (July 1974). "Formal requirements for virtualizable third generation architectures". Communications of the ACM. 17 (7): 412–421. doi:10.1145/361011.361073. S2CID 12680060
- [3] Selome Kostentinos Tesfatsion, Cristian Klein, and Johan Tordsson. 2018. Virtualization Techniques Compared: Performance, Resource, and Power Usage Overheads in Clouds ACM/SPEC International Conference on Performance Engineering (ICPE '18). https://doi.org/10.1145/3184407.3184414
- [4] https://www.redhat.com/es/topics/containers/containers-vs-vms