Name : Rohit R. Yakkundi
SRN : PES2UG23CS819
Section : G

Date 21-2-25 Page

1. In cloud computing, virtualization allows multiple (VMs) to run on a single physical machine by abstracting hardware resource.

Two main types : Full virtualization & Para virtualization.

Full virtualization :- It completely abstracts the hardware, allowing multiple OS to run on the same System. The hypervisor creates a complete virtual replica of the physical hardware.

Para-virtualization :- Para-virtualization requires modifications to the guest OS to communicate efficiently with hypervisor.

Working : Instead of emulating the full hardware exposes an interface that the guest OS can use diretly, reducing overhead.

Eg :- Xen, VMware, ESX (Older versions)

Campasison :

	Full-virtualization	Para-virtualization
OS Modification	Not required	Required
Performance	Lower	Higher
Hardware Emulation	Yes	Partial
Isolation	strong	Moderate.

2. Full virtualization Example Architecture:

- VMware ESXi, Microsoft Hyper-V, and KVM (Kernel-based-VM) are common examples of full virtualization architectures,
- In these architecture, the hypevisor completely abstracts the underlying hardware, allowing unmodified guest operating System to run in VM without awareness of being virtualized.
- ~~The~~ Para-virtualization Example Architecture:
- Xen (Para-virtualization Mode) is a well-known example of para-virtualization.
- In this approach, the guest OS is modified to be aware that it is running in a virtualized environment.
- Instead of executing privileged instructions directly, the guest OS makes explicit hypercalls to the hypervisor, reducing the need of complex instruction trapping and emulation.
- Changes/Modifications in Para-virtualization :-

1. Guest OS modification: The guest OS must be modified to include hypercalls instead of executing privileged CPU instructions directly.
2. No need for Binary Translation: Since the OS is aware of virtualization, the hypervisor doesn't need to use binary translation to handle sensitive instructions.

3. Efficient I/O Handling : The OS uses direct hypercalls for I/O operations, reducing the overhead compared to device emulation in full-virtualization.

3]

WSL 1 (windows Subsystem for Linux 1)

- Not a Virtual Machine nor a traditional container;
- WSL 1 translates Linux System calls into windows NT kernel calls using a compatibility layer.

WSL 2 (windows Subsystem for Linux 2)

- Uses a lightweight Virtual Machine (VM)
- Runs a real Linux Kernel inside a Hyper-V-based VM.
- Unlike traditional VMs, WSL 2 is optimized for fast boot times and lower resource usage, but still a VM at its core.

Conclusion :

- WSL 1 is not a VM / container it uses system call translation.
- WSL 2 is a VM, thought it is lightweight & optimized for performance.

4] WSL 2 : Full virtualization or para-virtualization ?-

WSL 2 uses Full virtualization

Justification :-

1) Runs an Unmodified Linux Kernel — unlike para-virtualization, WSL2 doesn't modify the Linux Kernel.

2) Uses Hardware virtualization — WSL 2 relies on Hyper-V full OS emulation, rather than modifying the guest OS for direct interaction.

3) No Hypercalls — para-virtualization requires hypercalls instead of privileged instruction but WSL2 doesn't modify Linux for th

4) Complete OS Isolation — The Linux Kernel in WSL2 run as a fully virtualized system, separate from windows unlike para-virtualized system.

5] Advantages of WSL 2 over virtual Box

1) Better Performance - WSL 2 runs a lightweight VM with faster file I/O and lower resource ~~ex~~ Usage.

2) Seamless Windows Integration - Direct file sharing & execution of linux command without extra setup.

3) Lower Resource Consumption - Uses less CPU & RAM compared to virtualBox, which req dedicated allocation.

4) Faster Boot time - Starts almost instantly unlike VB VMs that take longer to load.

5) Native GUI & Terminal Support - Supports linux GUI apps (WS and windows Terminal without extra configuration.

Conclusion:

WSL 2 is faster, more efficient, and better integrated with ~~with~~ windows compared to virtual Box.

6) Yes, the changes made to the Linux Kernel for WSL 2 are publicly available.

Discussion :-

- Open Source - Microsoft has made the WSL 2 Linux Kernel Source code available on Github under The GPLv2 license.

- Customer Optimizations - The Kernel is modified for Hyper-V compatibility, including optimizations for performance, memory management & system calls.

- Continuous Updates - Microsoft maintains & updates this Kernel, and users can build their own custom version if needed.

Conclusion :-

WSL'2's Linux Kernel is ~~OS~~ Open-source and available on GitHub, ensure transparency and allowing customization

7] WSL2 and KVM both use hardware-assisted virtualization, but they differ in purpose and implement. WSL 2 run a lightweight Hyper-V-based Vitual Machine to provide a Linux environment within windows,

Opetimized for seamless integration and low resource usage, It does not offer full VM management capabilities like traditional hypervisors. In contrast, KVM (Kernel-based Virtual Machine) is a full-fledged Linux hypervisor, converting the Linux Kernel into a Type-1 hypervisor, allowing multiple fully isolated VM's to run with near-native performance.