IMPLEMENTATION LEVELS OF VIRTUALIZATION



- •Virtualization is a computer architecture technology by which multiple virtual machines (vm) are multiplexed in the same hardware machine.
- •The purpose of a vm is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.



•Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system and software libraries) can be virtualized in various functional layers.

IMPLEMENTATION LEVELS OF

TITOMITAT TO A MICANI

Application level

JVM / .NET CLR / Panot

Library (user-level API) level

WINE/ WABI/ LxRun / Visual MainWin / vCUDA

Operating system level

Jail / Virtual Environment / Ensim's VPS / FVM

Hardware abstraction layer (HAL) level

VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux

Instruction set architecture (ISA) level

Bochs / Crusoe / QEMU / BIRD / Dynamo



INSTRUCTION SET ARCHITECTURE LEVEL

- •At the ISA level, virtualization is performed by **emulating a given ISA by the ISA of the host machine**. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.
- •Instruction set emulation leads to **virtual ISA** created on any hardware machine.

- - •The basic emulation method is through code interpretation
 - •An interpreter program interprets the source instructions to target instructions one by one.
 - •One source instruction may require tens or hundreds of native target instructions to perform its function.
 - Instruction set emulation requires binary translation and optimization.



HARDWARE ABSTRACTION LEVEL

- Hardware-level virtualization is performed right on top of the **bare hardware**. This approach generates a **virtual hardware environment** for a VM.
- ☐ The idea is to virtualize a **computer's resources**, such as its **processors, memory, and I/O devices**.
- ☐ The intention is to upgrade the hardware utilization rate by multiple users concurrently.



- •The idea was implemented in the IBM VM/370 in the 1960s.
- •The xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS applications.



OPERATING SYSTEM LEVEL

- This refers to an abstraction layer between traditional OS and user applications.
- OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers.
- The containers behave like real servers.



•OS-level virtualization is commonly used in **creating virtual hosting environments** to allocate hardware resources among a large number of **mutually distrusting users**.

• LIBRARY SUPPORT LEVEL

- Most applications use API exported by user-level libraries rather than using lengthy system calls by the OS.
- ❖ Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks. The software tool WINE has implemented this approach to support windows applications on top of UNIX hosts.



•Another example is the **VCUDA** Which allows applications executing within VMS **To leverage GPU Hardware** acceleration.



USER-APPLICATION LEVEL

- Virtualization at the application level virtualizes an application as a VM.
- On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization.
- The most popular approach is to deploy high level language (HLL) VMs.



- •Virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
- •The Microsoft .NET CLR and Java virtual machine (JVM) are two good examples of this class of VM.

RELATIVE MERITS OF VIRTUALIZATION AT VARIOUS LEVELS

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	Χ	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX