TAXONOMY OF VIRTUALIZATION TECHNIQUES

Virtualization refers to the use of various techniques that allow one system to mimic or emulate other systems or services. These techniques are applied in many areas of computing, providing flexibility, efficiency, and scalability. The classification of these virtualization techniques helps to better understand their characteristics and usage in different scenarios.

Virtualization Areas

Emulation Targets Virtualization techniques are mainly applied to emulate different services or entities. Specifically, virtualization is used to emulate three primary areas:

- **Execution Environments:** These are environments where programs, including operating systems and applications, can run as if they were on actual physical hardware.

- **Storage:** Virtualizing storage separates physical storage locations from the logical view users or programs interact with.
- **Networks:** Virtualizing networks allows different networks to operate as though they are independent of the underlying physical infrastructure.

Among these, execution virtualization is the oldest, most popular, and most developed area, making it the focus of most virtualization studies. Therefore, this area requires a deeper investigation and further categorization.

How is it Done?

1.Process-Level Virtualization

Process-level virtualization allows virtualization within the execution environment. This form of virtualization is hosted on an existing operating system and provides a virtualized process space where applications can run.

The main techniques used here include:

- **Emulation:** This technique mimics a different system entirely, allowing applications meant for one system to run on another by imitating its architecture.
- **High-Level Virtual Machines (VMs):** These VMs operate at a high level and abstract the details of the hardware. They are particularly suited for programming languages like Java, where the code runs on a virtual machine instead of directly on hardware.
- **Multiprogramming:** This involves running multiple processes on a single system in a time-sharing manner, where the OS switches between different processes. This is an example of virtualization at the operating system level.

2. System-Level Virtualization

System-level virtualization operates directly on the hardware without needing an existing operating system to host it. It is more efficient in accessing the hardware resources.

Key techniques under system-level virtualization are:

- **Hardware-Assisted Virtualization:** This technique uses the hardware's support to improve virtualization performance. It often involves using a hypervisor that interacts directly with the hardware to manage virtual machines.
- **Full Virtualization:** Here, the guest operating system runs in isolation as though it were directly using the hardware, without needing modification.
- **Paravirtualization:** In contrast to full virtualization, paravirtualization requires modification of the guest operating system so that it can interact more efficiently with the virtual environment.
- **Partial Virtualization:** Only part of the hardware resources are virtualized, meaning some applications may need to run directly on the host.

Virtualization Models

The virtualization models describe the environment created by the virtualization techniques:

Application-Level Virtualization

This model allows individual applications to run in a virtual environment separate from the underlying system. This is commonly achieved through emulation.

Programming Language-Level Virtualization

Programming language-level virtualization makes it possible for applications written in a particular language to run in a virtual machine designed for that language. High-level VMs such as the Java Virtual Machine (JVM) are examples of this type of virtualization model.

Operating System-Level Virtualization

This model allows multiple user environments to run on a single OS, managed by the multiprogramming technique. It ensures that resources are shared but isolated between environments.

Hardware-Level Virtualization

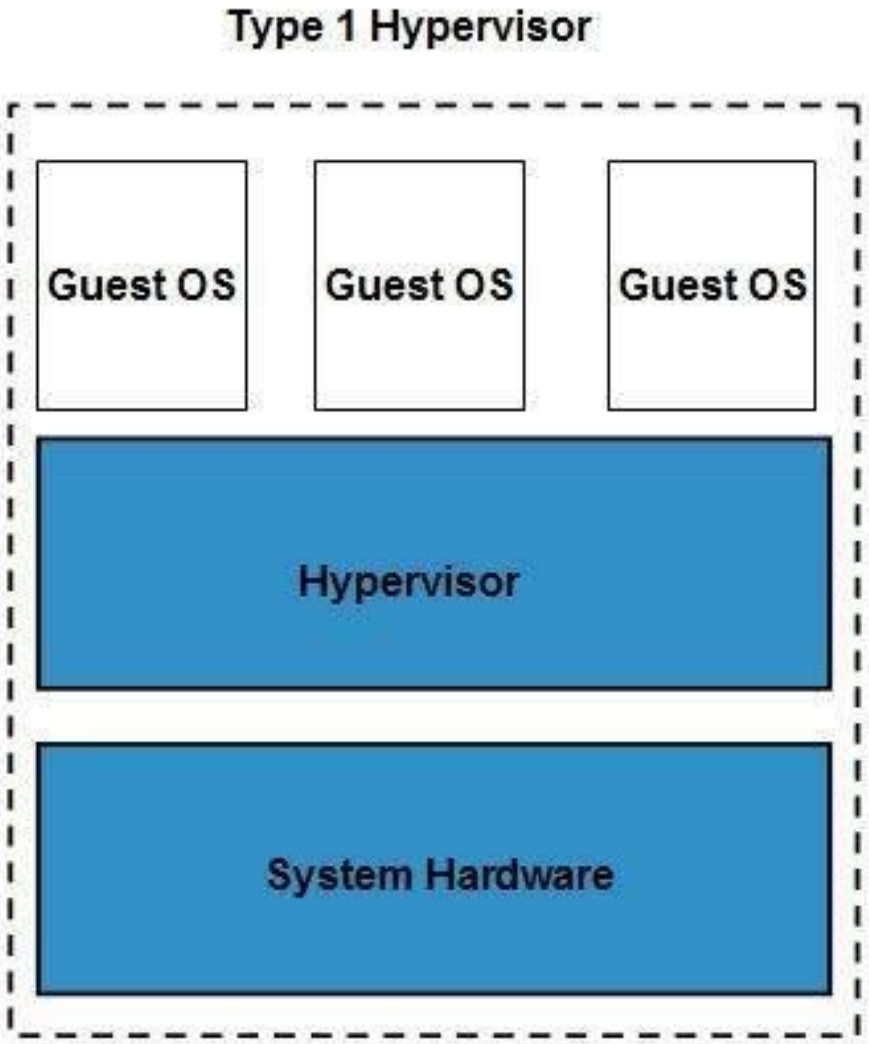
In system-level virtualization, hardware-level models are prevalent. These models allow multiple operating systems to share a single hardware platform through full, paravirtualization, or partial virtualization techniques.

Hypervisors

A hypervisor, also known as a Virtual Machine Manager (VMM), is essential in hardware virtualization. The hypervisor creates a virtual hardware environment where guest operating systems are installed. There are two types of hypervisors:

Type I Hypervisors:

These run directly on top of the hardware, meaning they replace the operating system. This type is known as a native virtual machine since it directly manages the hardware without needing an OS.



Type II Hypervisors:

These need the support of an existing operating system. The hypervisor runs as an application on the OS and interacts with the hardware for the guest systems. This is called a hosted virtual machine because the hypervisor relies on the host OS.

Type 2 Hypervisor