VM – Virtual Machine

> *This is a post from another blog of mine, which I'm shutting down. I like the way these virtualization concepts were worded in such a relatively simple way, so I'm keeping the post*

Virtual machine (VM) is a software implementation of a machine (computer) that executes programs like a real machine.

Two categories of virtual machines

Virtual machines are separated in two major categories, based on their use and degree of correspondence to any real machine.

A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS).

In contrast, a process virtual machine is designed to run a single program, which means that it supports a single process. An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine – it cannot break out of its virtual world.

System Virtual Machines

System virtual machines (sometimes called hardware virtual machines) allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. The software layer providing the virtualization is called a virtual machine monitor or hypervisor. A hypervisor can run on bare hardware (Type 1 or native VM) or on top of an operating system (Type 2 or hosted VM).

Main advantages of system VMs

- multiple OS environments can co-exist on the same computer, in strong isolation from each other;
- the virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine.

Main disadvantages of system VMs

- there’s still an overhead of the virtualization solution which is used to run and manage a VM, so performance of a VM will be somewhat slower compared to a physical system with comparable configuration
- virtualization means decoupling from physical hardware available to the host PC, this usually means access to devices needs to go through the virtualization solution and this may not always be possible

Multiple VMs each running their own operating system (called guest operating system) are frequently used in server consolidation, where different services that used to run on individual machines in order to avoid interference are instead run in separate VMs on the same physical machine. This use is frequently called quality-of-service isolation (QoS isolation).

Process Virtual Machines

A process VM, sometimes called an application virtual machine, runs as a normal application inside an OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform.

A process VM provides a high-level abstraction – that of a high-level programming language (compared to the low-level ISA

abstraction of the system VM). Process VMs are implemented using an interpreter; performance comparable to compiled programming languages is achieved by the use of just-in-time compilation.

This type of VM has become popular with the Java programming language, which is implemented using the Java virtual machine. Another example is the .NET Framework, which runs on a VM called the Common Language Runtime.

See Also

- VirtualBox
- Hardware Virtualization