Virtualisation

Table of Contents

Motivation

- **virtual machine**: efficient, isolated duplicate of a real machine
- **Virtual machine monitor (VMM)/Hypervisor**: piece of software that creates illusion of multiple virtual machines on the same physical hardware
- **virtualisation**: allows single computer to host multiple virtual machines
- advantages:
  - failure in one virtual machine doesn't bring down the system
  - can run multiple operating systems on the same hardware
  - fewer physical machines: less capital expenditure and operating costs
  - easier maintenance
  - ability to run legacy applications unsupported by current hardware
  - ability to test application in variety of environments

History

- seminal work 1974: Formal Requirements for Virtualizable Third Generation Architectures
  - listed conditions a computer should satisfy to support virtualisation efficiently
  - x86 didn't meet these requirements until 2005

Requirements for Virtualisation

Hypervisors should provide:

- **safety**: hypervisor should have full control of virtualised resources
- **fidelity**: behaviour of program on VM should be identical to behaviour on bare hardware
  - **sensitive instructions**: behave differently when executed in kernel mode c.f. user mode

  - **privileged instructions**: cause a trap if executed in user mode
  - a machine is virtualisable only if the sensitive instructions are a subset of the privileged instructions
  - i.e. if you try to do something in user mode that you should not be doing, the hardware should trap
  - Intel 386 didn't do this meaning couldn't support hypervisor directly

- **efficiency**: a substantial subset of the virtual processor's instructions should be executed directly by the real processor with no software intervention by the VMM
- **paravirtualisation**: presents a machine-like software interface, exposing the fact that it is a virtualised environment

  - provides **hypercalls**, allowing explicit requests to be sent to hypervisor
  - guests use hypercalls for privileged sensitive operations e.g. updating the page tables
  - by cooperating with hypervisor explicitly you get simpler and faster system

- **process-level virtualisation**: allow a process to run where the program was intended to run on a different architecture/OS

Type 1 and Type 2 Hypervisors

- terminology from Goldberg (1972)
- both pretend to be a full computer, and must execute machine's instruction set in a safe manner
- **type 1**: similar to an OS, as it is the only program running in most privileged mode

  - supports multiple copies of actual hardware (virtual machines) similar to processes an OS runs

- **type 2**: program that relies on an OS to allocate/schedule resources, similar to a regular process
- **guest operating system**: OS running on top of hypervisor
- **host operating system**: OS running on hardware (for Type 2 hypervisor)

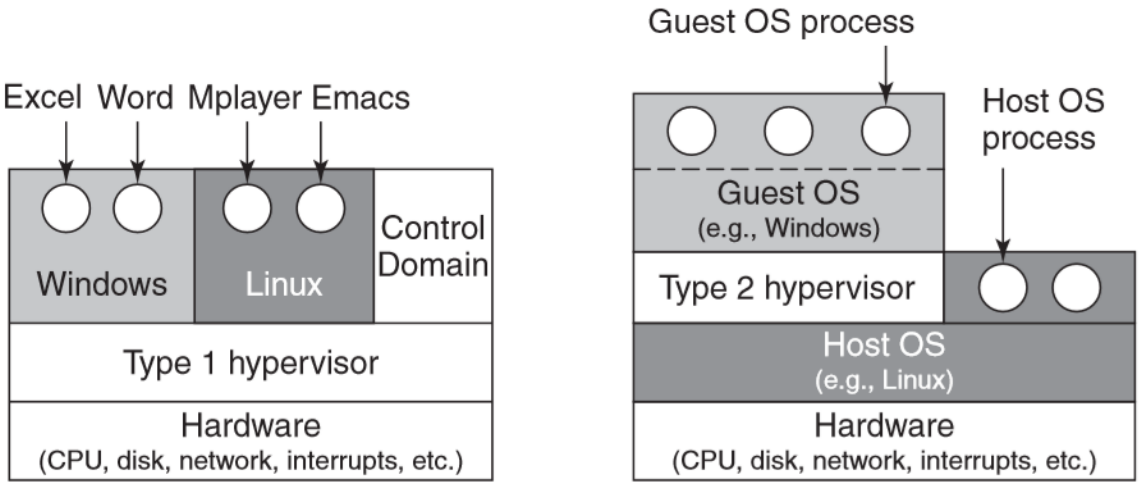

Figure 7-1. Location of type 1 and type 2 hypervisors.

Figure 1: hypervisor-types