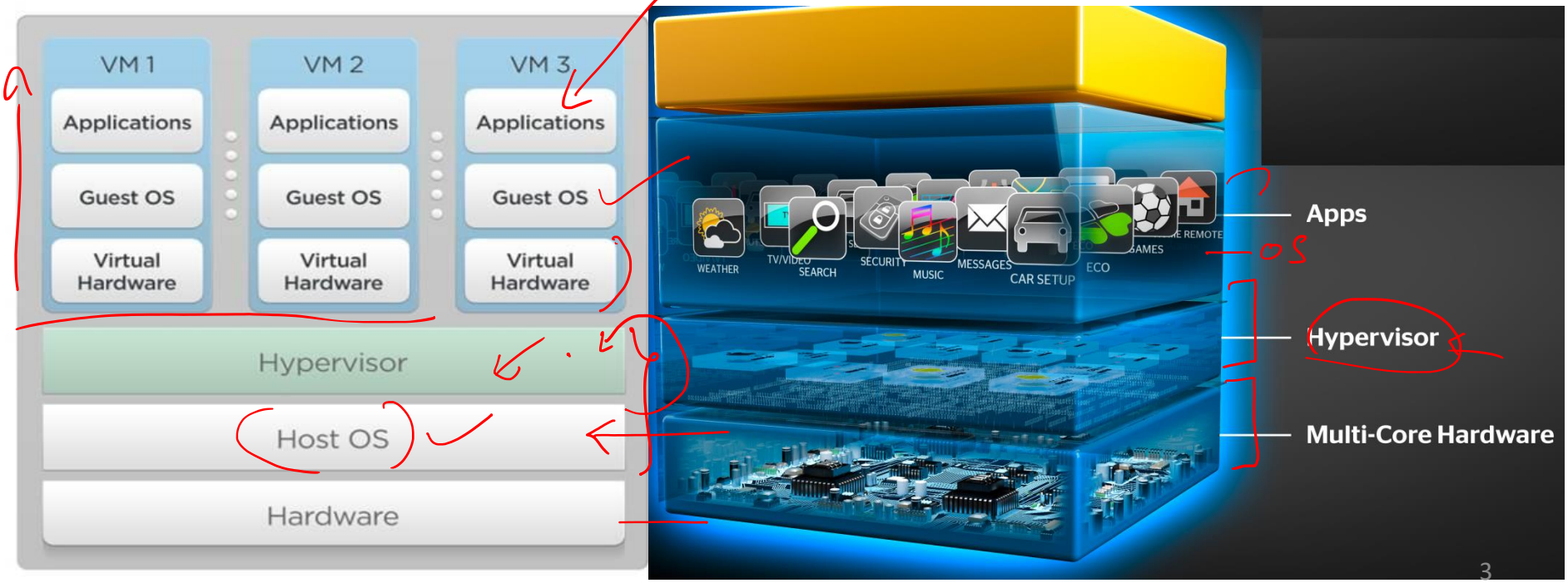
Cloud Computing

SEWP ZG527

BITS Pilani

Hypervisor

A thin layer of software that generally provides virtual partitioning capabilities which runs directly on hardware, but underneath higher-level virtualization services. Sometimes referred to as a “bare metal” approach.



Hypervisor Design Goals

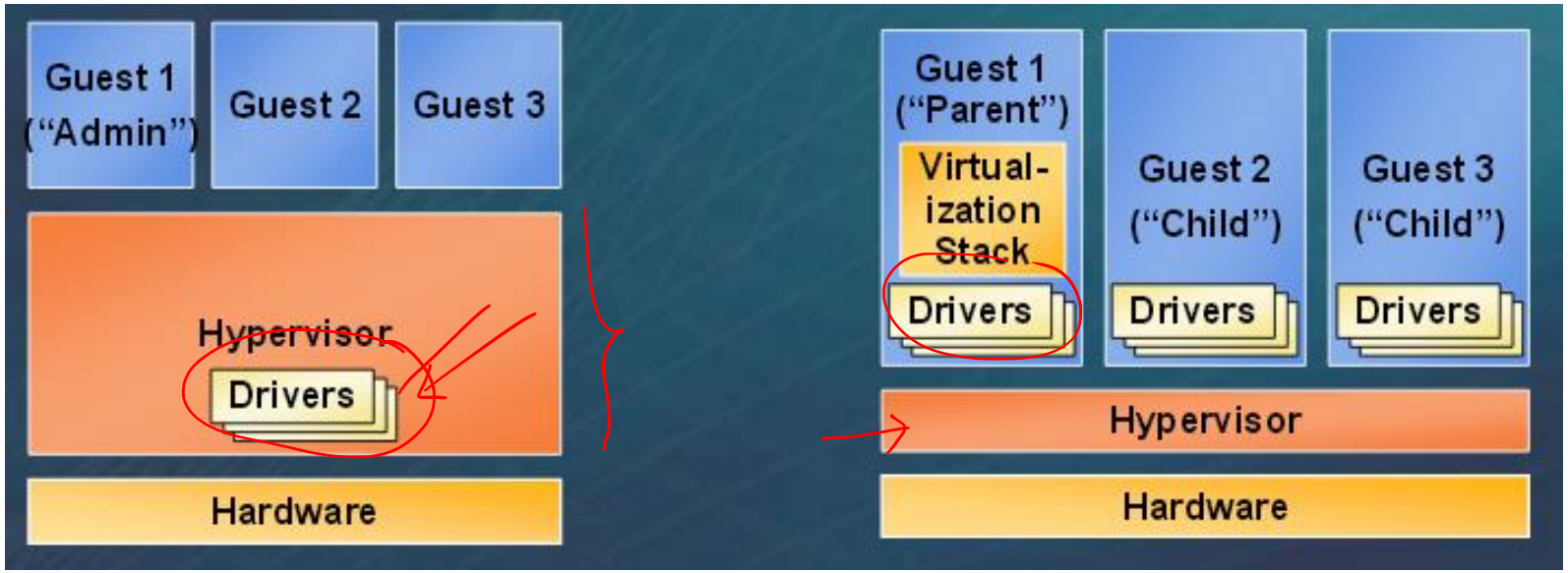
- Isolation
  - Security isolation
  - Fault isolation
  - Resource isolation
- Reliability
  - Minimal code base
  - Strictly layered design
  - Not extensible
- Scalability
  - Scale to large number of cores
  - Large memory systems

How Hypervisor goals are achieved?

- Partitioning Kernel
  - “Partition” is isolation boundary
  - Few virtualization functions; relies on virtualization stack
- Very thin layer of software
  - Microkernel
  - Highly reliable
  - Basis for smaller Trusted Computing Base (TCB)
- No device drivers
  - Drivers run in a partition
- Well-defined interface
  - Allow others to create support for their OSes as guests

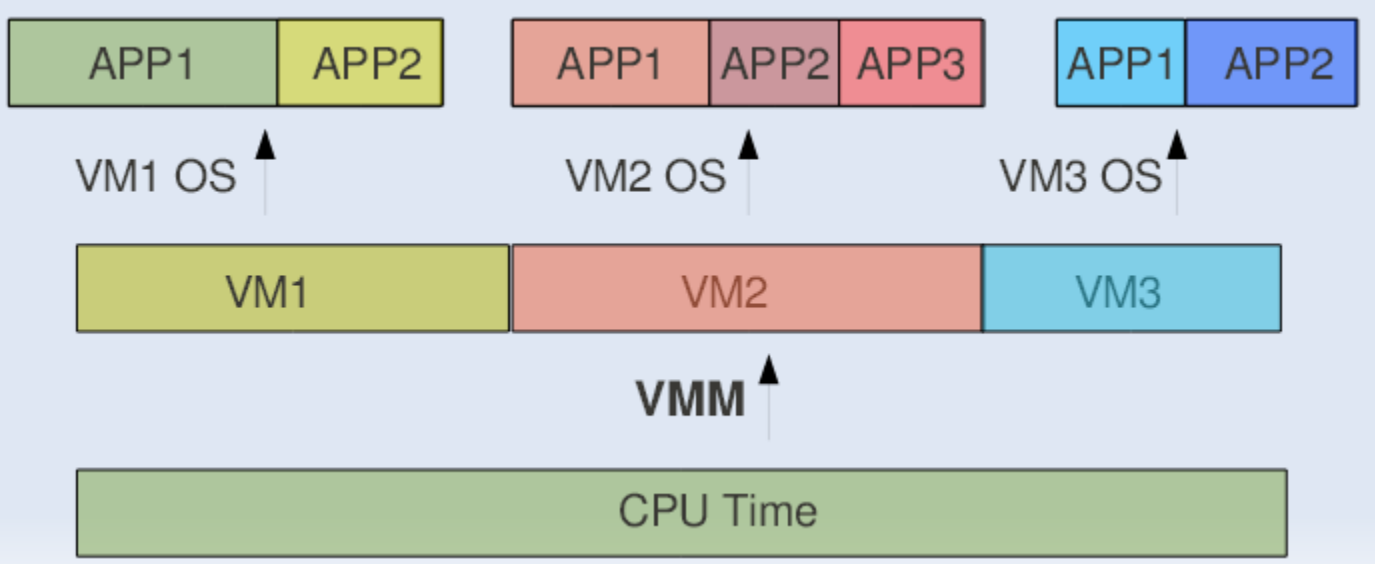
Hypervisor
Monolithic versus Microkernelized

- Monolithic hypervisor
  - Simpler than a modern kernel, but still complex
  - Contains its own drivers model
- Microkernelized hypervisor
  - Simple partitioning functionality
  - Increase reliability and minimize lowest level of the TCB
  - No third-party code
  - Drivers run within guests



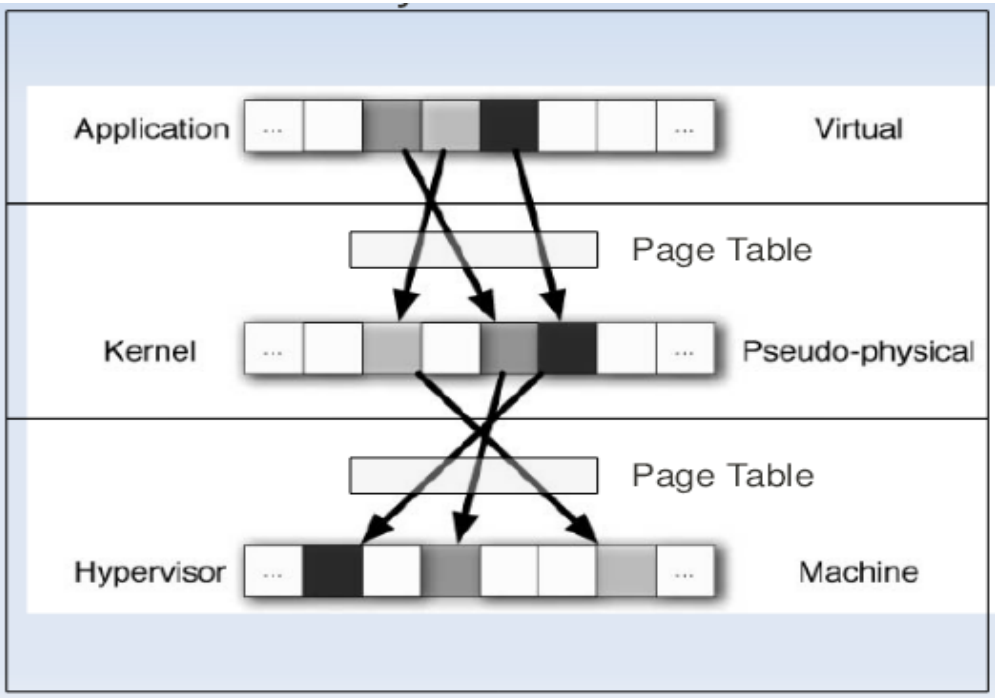
CPU Sharing

- VMM or Hypervisor provides a virtual view of CPU to VMs.
- In multi processing, CPU is alloted to the different processes in form of time slices by the OS.
- Similarly VMM or Hypervisor allots CPU to different VMs.



Memory Sharing

- In Multiprogramming there is a single level of indirection maintained by Kernel.
- In case of Virtual Machines there is one more level of indirection maintained by VMM



Applications use Virtual Addresses

Kernel translates Virtual Addresses to Pseudo-Physical Addresses

Hypervisor translates Pseudo-Physical Addresses to Machine addresses

IO Sharing

- Device needs to use Physical Memory location.
- In a virtualized environment, the kernel is running in a hypervisor-provided virtual address space
- Allowing the guest kernel to convey an arbitrary location to device for writing is a serious security hole
- Each device defines its own protocol for talking to drivers

Thanks!!!

Queries?