Virtual Machines

Readings: Chapter 14

Reasons for virtualization

- Legacy hardware
- Rapid deployment
- Versatility
- Consolidation
- Aggregating
- Dynamics
- Ease of management
- Increased availability

Virtual Machines

- A Virtual Machine is a software construct that mimics the characteristics of a physical server
- A VM instance is defined in files

- The principal functions performed by a hypervisor are:
  - Execution management of VMs
  - Devices emulation and access control
  - Execution of privileged operations by hypervisor for guest VMs
  - Management of VMs (also called VM lifecycle management)
  - Administration of hypervisor platform and hypervisor software

© 2017 Pearson Education, Inc., Hoboken, NJ. All rights reserved.
Container Virtualization

Container virtualization is a relatively recent approach to virtualization. In this approach, software, known as a virtualization container, runs on top of the host OS kernel and provides an isolated execution environment for applications.

Unlike hypervisor-based VMs, containers do not aim to emulate physical servers; instead, all containerized applications on a host share a common OS kernel. This eliminates the resources needed to run a separate OS for each application and can greatly reduce overhead.
Figure 14.5 Data Flow for I/O Operation via Hypervisor and Container

**Processor Issues**

- In a virtual environment there are two main strategies for providing processor resources:
  - Emulate a chip as software and provide access to that resource
  - Examples of this method are QEMU and the Android Emulator in the Android SDK
  - Provide segments of processing time on the physical processors (pCPUs) of the virtualization host to the virtual processors of the virtual machines hosted on the physical server
  - This is how most of the virtualization hypervisors offer processor resources to their guests

- Since hypervisor manages page sharing, the virtual machine operating systems are unaware of what is happening in the physical system

  - Ballooning
    - The hypervisor activates a balloon driver that (virtually) inflates and presses the guest operating system to flush pages to disk
    - Once the pages are cleared, the balloon driver deflates and the hypervisor can use the physical memory for other VMs

  - Memory overcommit
    - The capability to allocate more memory than physically exists on a host

- An advantage of virtualizing the workload's I/O path enables hardware independence by abstracting vendor-specific drivers to more generalized versions that run on the hypervisor
- This abstraction enables:
  - Live migration, which is one of virtualization's greatest availability strengths
  - The sharing of aggregate resources, such as network paths

- The memory overcommit capability is another benefit of virtualizing the I/O of a VM
- The trade-off for this is that the hypervisor is managing all the traffic and requires processor overhead

  - This was an issue in the early days of virtualization but now faster multicore processors and sophisticated hypervisors have addressed this concern

**Memory Management**

- Since hypervisor manages page sharing, the virtual machine operating systems are unaware of what is happening in the physical system

**I/O Management**

- An advantage of virtualizing the workload's I/O path enables hardware independence by abstracting vendor-specific drivers to more generalized versions that run on the hypervisor
- This abstraction enables:
  - Live migration, which is one of virtualization's greatest availability strengths
  - The sharing of aggregate resources, such as network paths

- The memory overcommit capability is another benefit of virtualizing the I/O of a VM
- The trade-off for this is that the hypervisor is managing all the traffic and requires processor overhead

  - This was an issue in the early days of virtualization but now faster multicore processors and sophisticated hypervisors have addressed this concern