Operating Systems

Virtualization

Carlos Alberto Llano R.

20 de octubre de 2015

Contents

Overview

The fundamental idea behind a virtual machine is to abstract the hardware of a single computer (the CPU, memory, disk drives, network interface cards, and so forth) into several different execution environments, thereby creating the illusion that each separate environment is running on its own private computer.

Overview - Components

host it's a physical machine.

hypervisor or Virtual Machine Manager, it's runs the virtual machines.

guest it's the virtual machine.

Overview - The implementation of VMMs

- **Hardware-based solutions via firmware**, these are generally known as type 0 hypervisors.
- **Operating-system-like software built to provide virtualization**: VMware ESX(mentioned above), Joyent SmartOS, and Citrix XenServer, these are known as type 1 hypervisors.
- **General-purpose operating systems**: Microsoft Windows Server with HyperV and RedHat Linux with the KVM feature.

Overview - The implementation of VMMs

- **Applications that run on standard operating systems**: VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox, are type 2 hypervisors.
- **Paravirtualization**, a technique in which the guest operating system is modified to work in cooperation with the VMM to optimize performance.
- **Programming-environment virtualization**, in which VMMs do not virtualize real hardware but instead create an optimized virtual system: Oracle Java and Microsoft.Net.

Overview - The implementation of VMMs

- **Emulators** that allow applications written for one hardware environment to run on a very different hardware environment, such as a different type of CPU.
- **Operating System Level Virtualization or containers**. Under this system, there is only one kernel installed - the host kernel. Each container is simply an isolation of the userland processes.

History

Virtual machines first appeared commercially on IBM mainframes in 1972.

Virtualization was provided by the IBM VM operating system.

A major difficulty with the VM approach involved disk systems. The solution was to provide virtual disks—termed minidisks in IBM's VM operating system.

Benefits and Features

- Ability to share the same hardware yet run several different execution environments concurrently.
- The host system is protected from the virtual machines, just as the virtual machines are protected from each other.
- Energy costs.
- Increases Business Agility.
- Increases IT Operational Flexibility.
- Reduces IT Operations Costs.
- High Availability.
- Disaster Recovery.
- Is Green.

Benefits and Features

- Suspend
- Snapshots
- Resume
- Clone
- Templating
- Live migration

Open Virtual Machine Format

The DMTF's Open Virtualization Format (OVF) standard provides the industry with a standard packaging format for software solutions based on virtual systems, solving critical business needs for software vendors and cloud computing service providers. OVF has been adopted and published by the International Organization for Standardization (ISO) as ISO 17203.

Cloud Computing

is made possible by virtualization in which resources such as CPU, memory, and I/O are provided as services to customers using Internet technologies. By using APIs, a program can tell a cloud computing facility to create thousands of VMs, all running a specific guest operating system and application, which others can access via the Internet.

Virtual CPU - VCPU

The VCPU does not execute code. Rather, it represents the state of the CPU as the guest machine believes it to be.

When the guest is context-switched onto a CPU by the VMM, information from the VCPU is used to load the right context, much as a general-purpose operating system would use the PCB.

VirtualBox

VirtualBox allows you to run practically any operating system right inside your current OS.

Installation:
https://help.ubuntu.com/community/VirtualBox/Installation
Ubuntu iso server: http://www.ubuntu.com/download/server

VirtualBox - Network types

Network Address Translation (NAT) means the virtual machines will have private IP addresses that are not routable from outside.

Example: Your host is 192.168.1.1. The VirtualBox NAT device will be marked as 10.0.2.1. Therefore, the virtual machines will be given any address in the 10.0.2.x range.

VirtualBox - Network types

Bridge Adapter means that any virtual machine running will try to obtain an IP address from the same source your currently active, default network address got its IP address.

Example: Your host has leased an address of 192.168.1.100 from the router. The virtual machine leases an address of 192.168.1.103 from the router. The two machines now share the same network and all standard rules apply. For all practical purposes, the virtual machine is another IP address on your LAN.

VirtualBox - Network types

Host-only Adapter It's very similar to Bridged Adapter, except that is uses a dedicated network device, called vboxnet0, to lease IP addresses.

Example: Your host has the IP address of 192.168.56.1. Your virtual machine has the IP address of 192.168.56.101.

VMware Server has its two virtual adapters called vmnet1 and vmnet8, which are used assign NAT and host-only IP addresses to guests.

VirtualBox - Network types

Internal network It's similar to Host-only + NAT, except the networking takes place inside the virtual network of guest machines, without any access for the host, plus there is no real NAT. What you get is a private LAN for your guests only, without any access to the external world.

VirtualBox - Ubuntu interfaces

ifconfig −a

In /etc/network/interfaces file:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface NAT
auto eth0
iface eth0 inet dhcp

# The seconday network interface  HOST−ONLY ADAPTER
auto eth1
iface eth1 inet static
address 192.168.56.103      <−−−−−−−−−−−−−−−−−−−−−−−− Ip of range of adapter
netmask 255.255.255.0
```

VirtualBox - Exercise

Install two ubuntu servers and to configurate the network.

In any machine create a snapshot after install apache, after retore the snapshot.

VirtualBox - Ubuntu interfaces

List all vms
==========
VBoxManage list vms
VBoxManage list ostypes

List all properties
==========
VBoxManage guestproperty enumerate testMachine

Create a new machine
==========
VBoxManage createvm −−name testMachine −−ostype Ubuntu_64 −−**register**
Output:
Virtual machine 'testMachine' is created and registered.
UUID: 8f368cc1−7f9e−4378−a0c3−1f84dffe87c8
Settings file: '/home/callanor/VirtualBox VMs/testMachine/testMachine.vbox'

Checking existing Virtual machine
==========
VBoxManage showvminfo testMachine

Change memory
==========
VBoxManage modifyvm testMachine −−memory 1024

VirtualBox - Ubuntu interfaces

Set cores

```
VBoxManage modifyvm testMachine --cpus 1 --ioapic on
```

Create a bridge adapter

```
VBoxManage modifyvm testMachine --bridgeadapter1 eth0
VBoxManage modifyvm testMachine --nic1 bridged
```

Create an HDD and attach

```
VBoxManage createhd --filename testMachine.vdi --size 18000 --format VDI
VBoxManage storagectl testMachine --name "SATA Controller" --add sata --controller IntelAhci
VBoxManage storageattach "testMachine" --storagectl "SATA Controller" --port 0 --device 0 --type
      hdd --medium testMachine.vdi
VBoxManage storagectl testMachine --name "IDE Controller" --add ide --controller PIIX4
VBoxManage storageattach testMachine --storagectl "IDE Controller" --port 1 --device 0 --type
      dvddrive --medium /tmp/ubuntu.iso
```

Attach VBoxGuestAdditions

```
VBoxManage storageattach testMachine --storagectl "IDE Controller" --port 1 --device 0 --type
      dvddrive --medium /usr/share/virtualbox/
VBoxGuestAdditions.iso
VBoxManage modifyvm $vm --dvd /usr/share/virtualbox/VBoxGuestAdditions.iso
```

Vagrant

Vagrant makes it really easy to work with virtual machines. According to the Vagrant docs.

Installation: http://www.vagrantup.com/downloads vagrant init ubuntu/trusty32; vagrant up –provider virtualbox vagrant ssh http://www.vagrantbox.es/

Trap and Emulate

An operation system is designed to have full control of system. But when an OS in running as a virtual machine in a hypervisor some of its instruction may conflict with the host operation system so what hypervisor do, it emulates the effect of that specific instruction or action without being carrying out it. In this way host OS is not effected by guest's actions. This is called trap and emulate.

Trap and Emulate