Virtualización (Capítulo 14) – Máquinas Virtuales

Curso: Sistemas Operativos II
Alumno: Alejandro Vargas

Este material resume los conceptos principales del Capítulo 14 sobre máquinas virtuales (partes 1 y 2). Se incluyen comandos y pequeños scripts.

Contenido

12. Bibliografía

---

1) Definición de virtualización

La virtualización consiste en una capa intermedia que permite ejecutar múltiples sistemas operativos invitados sobre un mismo equipo físico. El **hipervisor** o **monitor de máquinas virtuales (VMM)** administra recursos como procesador, memoria, disco y red.

- Favorece la consolidación de servicios en menos servidores físicos.
- Aísla fallos y simplifica pruebas y enseñanza.
- Permite tomar instantáneas, clonar y migrar VMs.

---

2) Clases de hipervisores

- **Tipo 1 (bare-metal):** se ejecutan directamente sobre el hardware (ejemplo: ESXi, Hyper-V bare-metal).
- **Tipo 2 (hosteados):** funcionan como aplicaciones dentro de un sistema anfitrión (ejemplo: VirtualBox, VMware Workstation).

En ambos casos, el VMM controla el acceso a instrucciones críticas y a dispositivos virtuales. Extensiones como Intel VT-x o AMD-V mejoran la eficiencia reduciendo el overhead.

---

3) Estrategias de virtualización

- **Virtualización completa:** emulación de hardware y traducción de instrucciones privilegiadas.
- **Paravirtualización:** el invitado coopera usando drivers o interfaces especiales (ejemplo: virtio).
- **Asistida por hardware:** el CPU integra modos especiales que permiten la ejecución directa de invitados.

---

4) Virtualización de recursos

CPU:

- Planificación por turnos entre vCPUs.
- Afinidad con núcleos físicos para mayor estabilidad.
- Overcommit moderado para densidad sin comprometer demasiado el rendimiento.

Memoria:

- RAM asignada por VM, con técnicas como ballooning y deduplicación (KSM).
- Precaución con el uso de swap en el host, que aumenta latencia.

Almacenamiento:

- Archivos de disco virtual (VDI, VMDK, QCOW2) con thin/thick provisioning.
- Cachés y colas de E/S influyen en rendimiento (IOPS/latencias).
- Snapshots son útiles pero consumen espacio y E/S.

Red:

- Configuraciones NAT, puente (bridged) y host-only.
- Uso de switches virtuales o bridges para comunicación entre VMs y red física.

---

5) Beneficios, restricciones y seguridad

- **Beneficios:** consolidación, aislamiento, rapidez de despliegue, snapshots, clonación, migración.
- **Limitaciones:** overhead, competencia por recursos (“vecino ruidoso”), gestión compleja.
- **Seguridad:** segmentación de redes, actualización de drivers Guest Additions/virtio, control de accesos.

---

6) Contenedores vs VMs

- Los contenedores comparten el kernel del anfitrión, mientras que las VMs traen su propio kernel.
- Los contenedores suelen ser más ligeros; las VMs brindan mayor aislamiento.
- Ambas tecnologías pueden usarse juntas (ejemplo: Kubernetes sobre VMs).

---

7) Operaciones comunes con VMs

- Creación y registro de VMs, adición de discos e ISOs.
- Configuración de redes (NAT, bridge, host-only).
- Toma y restauración de snapshots; clonación de máquinas base.
- Automatización mediante scripts o herramientas (ejemplo: Vagrant, cloud-init).

---

8) Prácticas rápidas

1. Crear una VM Linux mínima con 1–2 vCPU y 1–2 GB de RAM.
2. Instalar herramientas de evaluación: `sysbench` (CPU), `fio` (disco), `iperf3` (red).
3. Tomar un snapshot antes de pruebas y luego restaurarlo.
4. Medir desempeño (tiempos, IOPS, throughput).

---

9) Comandos útiles en VMs

CPU (sysbench):

```
sudo apt-get update && sudo apt-get install -y sysbench
sysbench cpu --cpu-max-prime=20000 run
```

Disco (fio):

```
sudo apt-get install -y fio
fio --name=seqwrite --rw=write --bs=1M --size=1G --iodepth=32
--direct=1 --filename=fiotest.img
fio --name=seqread --rw=read --bs=1M --size=1G --iodepth=32
--direct=1 --filename=fiotest.img
fio --name=randrw --rw=randrw --bs=4k --size=1G --iodepth=64
--direct=1 --filename=fiotest.img
```

Red (iperf3):
En la VM servidor:

```
sudo apt-get install -y iperf3
iperf3 -s
```

En la VM cliente:

```
iperf3 -c <IP_VM_SERVIDOR> -t 30
```

---

10) Scripts cortos de apoyo

10.1 VirtualBox (Bash):

```
#!/usr/bin/env bash
VM_NAME="SO2-Ubuntu-Mini"
ISO_PATH="/isos/ubuntu-24.04-live-server-amd64.iso"
RAM_MB=2048
VCPUS=2
DISK_MB=20000
VBOXM="VBoxManage"

$VBOXM createvm --name "$VM_NAME" --register
$VBOXM modifyvm "$VM_NAME" --memory $RAM_MB --cpus $VCPUS --firmware
efi --nic1 nat
$VBOXM storagectl "$VM_NAME" --name "SATA" --add sata --controller
IntelAhci
$VBOXM createmedium disk --filename "$HOME/VirtualBox
VMs/$VM_NAME/$VM_NAME.vdi" --size $DISK_MB
```

```
$VBOXM storageattach "$VM_NAME" --storagectl "SATA" --port 0
--device 0 --type hdd --medium "$HOME/VirtualBox
VMs/$VM_NAME/$VM_NAME.vdi"
$VBOXM storagectl "$VM_NAME" --name "IDE" --add ide
$VBOXM storageattach "$VM_NAME" --storagectl "IDE" --port 0 --device
0 --type dvddrive --medium "$ISO_PATH"
$VBOXM startvm "$VM_NAME" --type gui
```

10.2 Vagrantfile:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/jammy64"
  config.vm.provider "virtualbox" do |vb|
    vb.cpus = 2
    vb.memory = 2048
  end
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y sysbench fio iperf3
  SHELL
end
```

10.3 KVM/QEMU (Bash):

```
#!/usr/bin/env bash
set -e
VM="so2-mini"
RAM=2048
CPUS=2
DISK=/var/lib/libvirt/images/${VM}.qcow2
ISO=/isos/ubuntu-24.04-live-server-amd64.iso

qemu-img create -f qcow2 "$DISK" 20G
virt-install --name "$VM" --memory "$RAM" --vcpus "$CPUS" --disk
path="$DISK",format=qcow2 --cdrom "$ISO" --os-variant ubuntu24.04
--network network=default --graphics vnc --noautoconsole
```

10.4 Hyper-V (Bash simulado para creación de VM con parámetros):

```
#!/usr/bin/env bash
VMName="SO2-HyperV-Mini"
```

```
VhdPath="/HyperV/$VMName.vhdx"
Switch="Default Switch"

New-VHD -Path $VhdPath -SizeBytes 20GB -Dynamic
New-VM -Name $VMName -MemoryStartupBytes 2GB -Generation 2 -VHDPath
$VhdPath -SwitchName "$Switch"
Set-VMProcessor -VMName $VMName -Count 2
Add-VMDvdDrive -VMName $VMName -Path
"/isos/ubuntu-24.04-live-server-amd64.iso"
Start-VM $VMName
```

---

11) Recomendaciones de rendimiento

- Activar VT-x/AMD-V en BIOS/UEFI y mantener firmware actualizado.
- Evitar sobreasignar CPU/RAM en exceso.
- Usar drivers paravirtualizados (virtio) y mantener Guest Additions/Tools al día.
- Utilizar snapshots de forma estratégica y eliminarlos cuando ya no se necesiten.
- Separar redes de laboratorio y limitar accesos externos.

---

12) Bibliografía

- Videos Capítulo 14 – Máquinas virtuales (partes 1 y 2).
- Manual oficial de VirtualBox (VBoxManage).
- Documentación de KVM/QEMU (virt-install), Vagrant e Hyper-V.