Using KVM Virtualization

Narges Mehran, M.Sc.
Dr. Dragi Kimovski

Current Topics in Distributed Systems: Internet of Things and Cloud Computing,

SS21

Virtualization

- The technology to execute multiple virtual machines (VMs) on single server hardware,

- it reduces hardware expenses and helps minimize infrastructure costs such as power consumption and cooling,

- virtualization provides the operational flexibility required in today's service-oriented, high-availability IT operations by making it possible:
  - to migrate running VMs from one physical host to another when mandated by hardware or physical plant problems, or
  - to maximize performance through load balancing or in response to increasing processor and memory requirements.

Virtualization types

- Hardware Emulation (full virtualization)
  - A hypervisor presents an emulated hardware to unmodified guest operating systems.
  - Example: VMware desktop/server, VirtualBox, QEMU.
- Paravirtualization
  - A hypervisor multiplexes access to hardware by modified guest operating systems.
  - Example: Xen.
- A hardware assisted virtualization on re-designed x86 platforms, such as AMD-V and Intel-VT
  - Example: KVM, VirtualBox, VMware ESX, Hyper-V.

KVM: Kernel Virtual Machine

- An open-source virtualization technology built into Linux,
  - turns Linux into a hypervisor that allows virtual guests to be created,
  - uses Linux for process management such as memory management, io, scheduling,
  - included in kernel version 2.6.20 and later.

- KVM uses Virtualization extensions in the processor chips,
  - Intel: VT-x (or vmx),
  - AMD: AMD-V (or svm).

Why virtualization extensions?

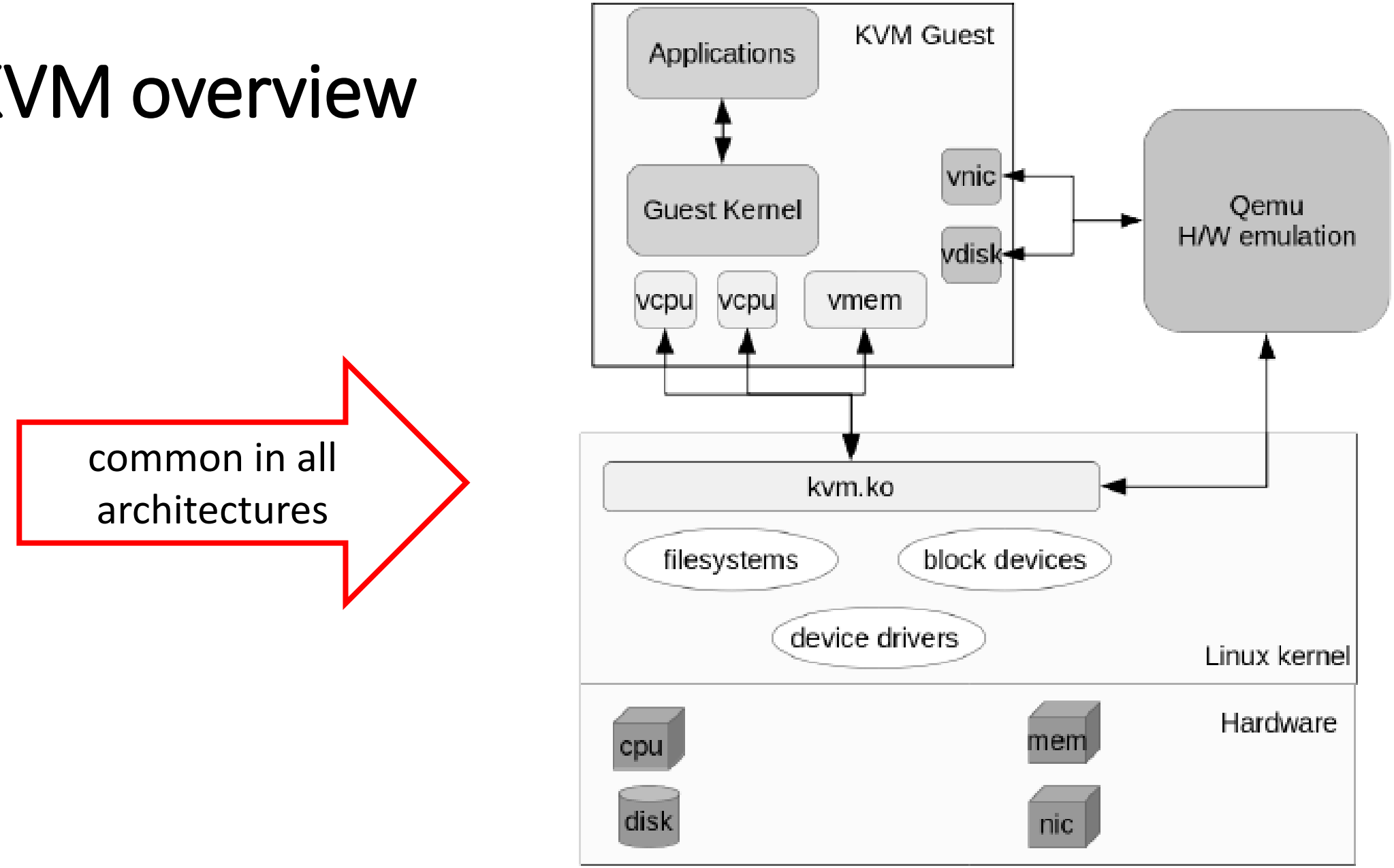
- When the hypervisor emulates a CPU, it translates the instructions related to vCPU to the physical CPU,
- this impacts on the performance,
- therefore, modern processors support virtualization extensions, such as Intel VT-x and AMD-V,
- these technologies provide the ability for a slice of the physical CPU to manage the vCPU,
- then, the instructions related to the vCPU will be run on the physical CPU slice.

Quick Emulator (QEMU)

- QEMU is a Hypervisor or VMM,
  - emulates the CPU
  - provides device characteristics
  - uses dynamic binary translation to work as an emulator
  - uses hardware virtualization extensions while working with KVM

- The combination of QEMU and KVM,
  - can achieve a good performance

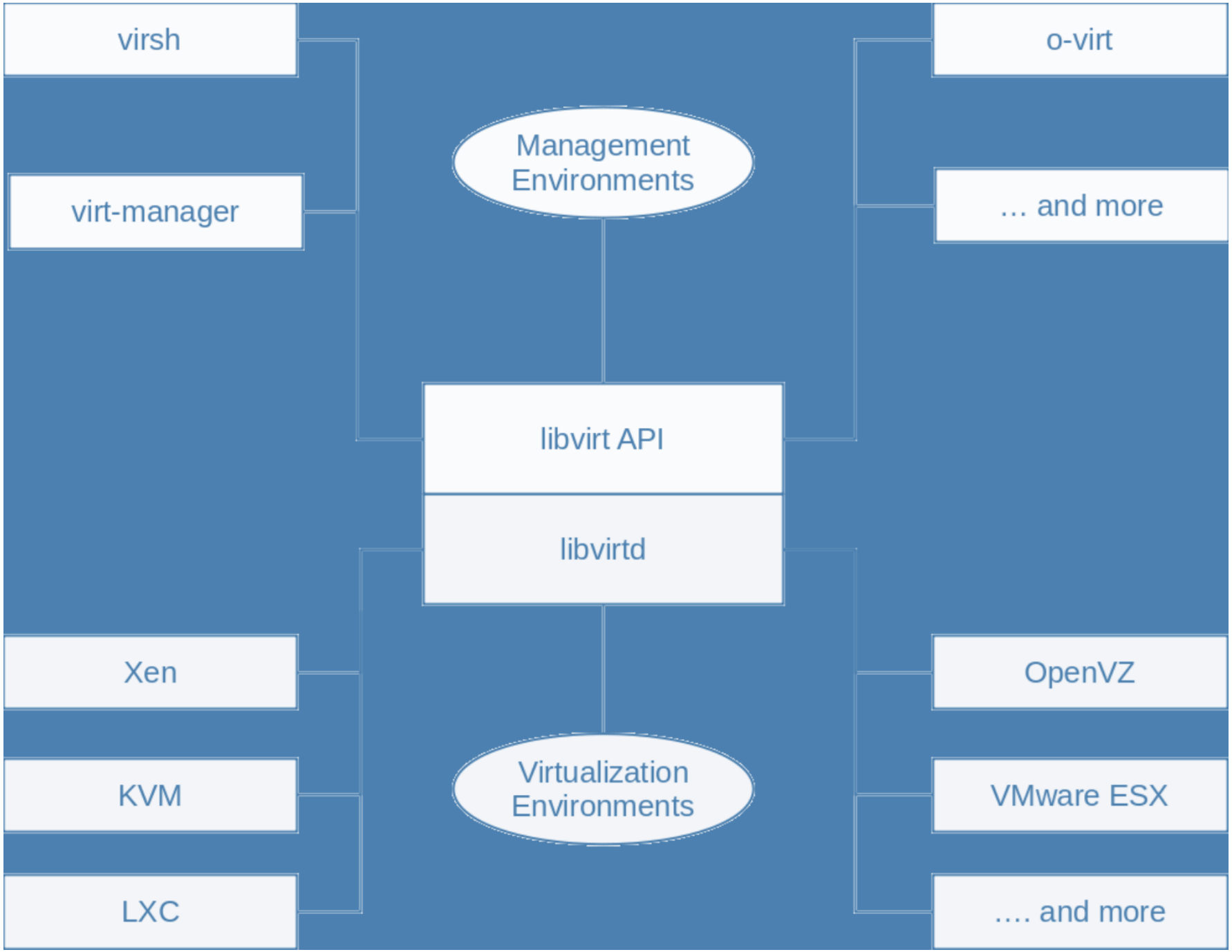
KVM overview



libvirt

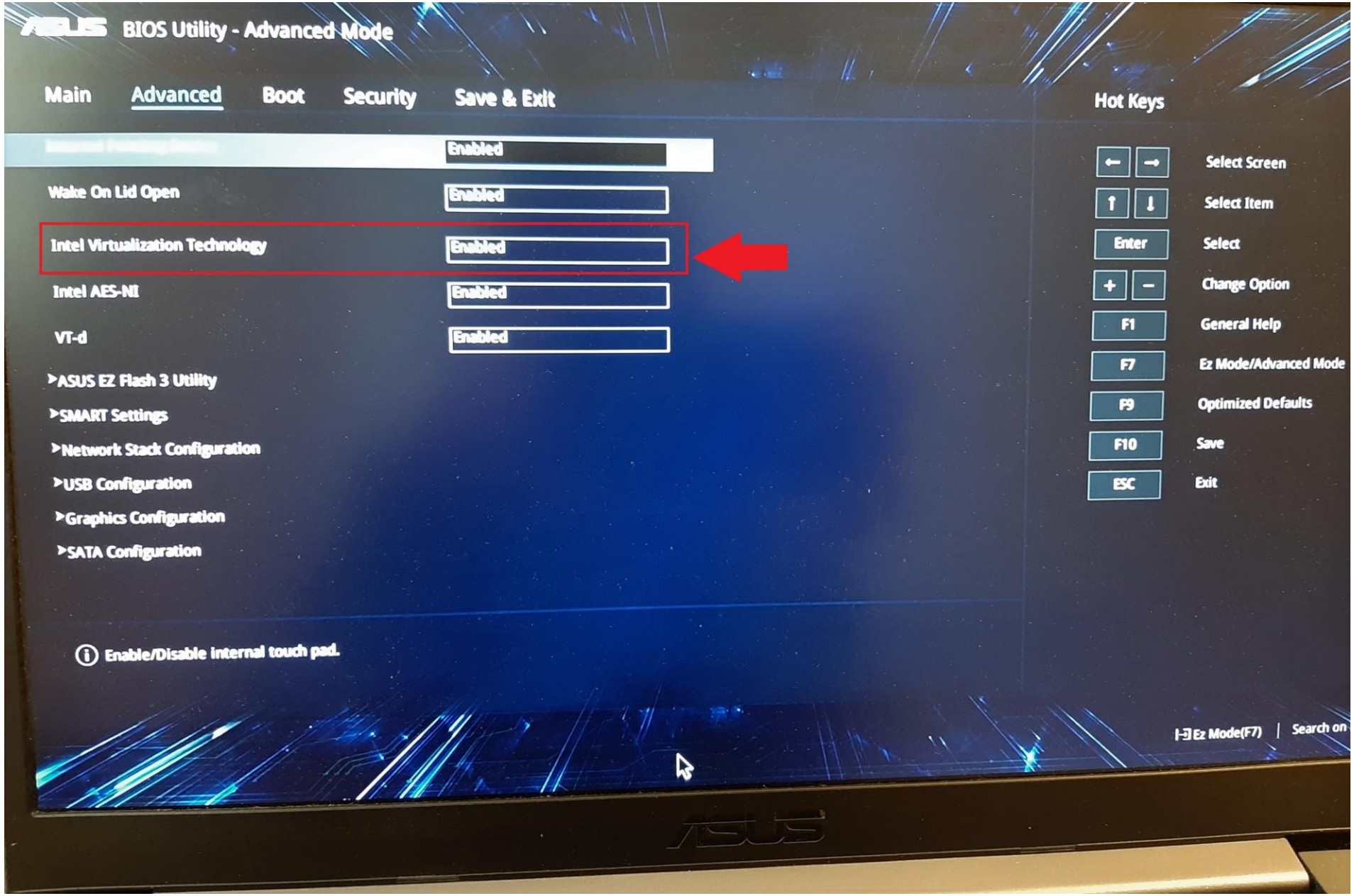
- libvirt:
  - is an open-source management toolkit for managing virtualization platforms, https://libvirt.org/

- The libvirt API supports virtualization technologies such as KVM, Xen, LXC containers, OpenVZ, User-mode Linux, VirtualBox, Microsoft® Hyper-V®, and many Vmware technologies.



Pre-installation checklist

- Please Check that your CPU supports hardware virtualization.
- How to check the support for hardware virtualization?
- To run KVM, you need a processor that supports hardware virtualization. Intel and AMD both have developed extensions for their processors:
  - Intel VT-x
  - and AMD-v
- To see if your processor supports one of these, you can review the output from this command:
  - `egrep -c '(vmx|svm)' /proc/cpuinfo`
    - If 0 it means that your CPU doesn't support hardware virtualization.
    - If 1 or more it does - but you still need to make sure that virtualization is enabled in the BIOS.


BIOS Utility - Advanced Mode
Main
Advanced
Boot
Security
Save & Exit
Hot Keys
Enabled
Wake On Lid Open
Enabled
Intel Virtualization Technology
Enabled
Intel AES-NI
Enabled
VT-d
Enabled
ASUS EZ Flash 3 Utility
SMART Settings
Network Stack Configuration
USB Configuration
Graphics Configuration
SATA Configuration
Select Screen
Select Item
Enter
Select
Change Option
F1
General Help
F7
Ez Mode/Advanced Mode
F9
Optimized Defaults
F10
Save
ESC
Exit
Enable/Disable internal touch pad.
Ez Mode(F7)
Search on

Introduction to virsh

- `virsh` provides a command-line interface to manage services provided by `libvirt`. The services that `virsh` can manage:
  - Guest Domains
  - Device
  - Virtual Network
  - Virtual interface
  - Storage Pool
- The `virsh` documentation: https://libvirt.org/virshcmdref.html

virsh installation

- To install, you need the following packages:
  - `sudo apt-get install qemu qemu-kvm libvirt-clients libvirt-daemon-system libvirt-bin libosinfo-bin acpid virt-manager virtinst bridge-utils`
  - `sudo systemctl enable libvirtd`
  - `sudo systemctl start libvirtd`

virsh: useful commands

- Define the virtual machine (by KVM XML file, you define all the resources that the KVM guest is going to use).
  - ✓ `virsh define mytiny.xml`
- verify the configuration file
  - ✓ `virsh dumpxml mytiny`
- start the virtual machine
  - ✓ `virsh start mytiny`
- connect to the virtual machine
  - ✓ `virt-viewer mytiny`
- exit the virt-viewer and verify the state of the VM
  - ✓ `virsh list --all`
- remove the libvirt configuration for VM
  - ✓ `virsh undefine mytiny`
- shutdown or force the VM to shut (destroy) a registered domain:
  - ✓ `virsh shutdown mytiny`
  - ✓ `virsh destroy mytiny`

KVM hardware accelerated guest virtual machine on i686

```
<domain type='kvm'>
  <name>demo2</name>
  <uuid>4dea24b3-1d52-d8f3-2516-782e98a23fa0</uuid>
  <memory>131072</memory>
  <vcpu>1</vcpu>
  <os>
    <type arch="i686">hvm</type>
  </os>
  <clock sync="localtime"/>
  <devices>
    <emulator>/usr/bin/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <source file='/var/lib/libvirt/images/demo2.img'/>
      <target dev='hda'/>
    </disk>
    <interface type='network'>
      <source network='default'/>
      <mac address='24:42:53:21:52:45'/>
    </interface>
    <graphics type='vnc' port='-1' keymap='de'/>
  </devices>
</domain>
```

Figure 20.71. Example domain XML config

Source of image: redhat.com

Libvirt API

- The helpful tools:
  - `virt-manager` graphic tool for creating and managing virtual machines, KVM, Xen and LXC.
    - Documentation:
      - https://virt-manager.org
  - `virt-viewer` graphic tool for connecting to the client VM, uses VNC or SPICE protocols
  - `virt-install` command driven installer for creating VM

virt-install

- `virt-install` is a command-line tool that allows you to create KVM guests in Linux:

```
virt-install \
   --name firsttest \
   --memory 1024 \
   --disk /home/narges/Downloads/ubuntu-16.04-desktop-i386.iso \
   --import
```

- You can download such as a distribution:
  - http://releases.ubuntu.com/releases/xenial/ubuntu-16.04.6-desktop-i386.iso

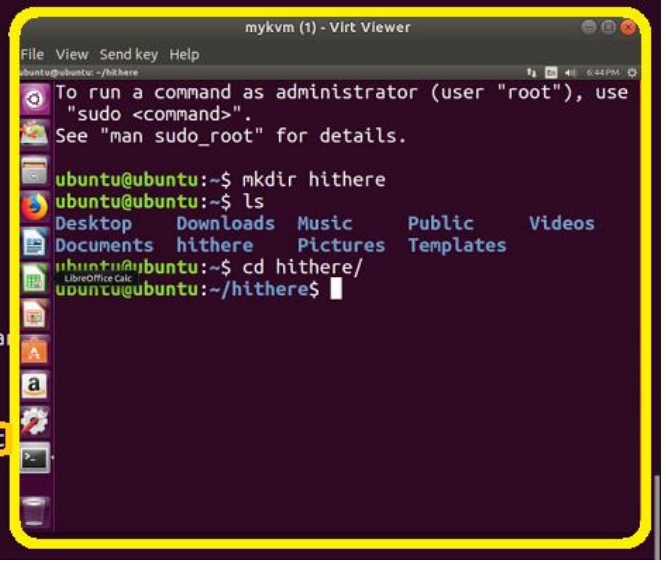
virt-install --import

- The option import to the `virt-install` command:
  - ✓skips the install process
  - ✓creates a default configuration for the new VM
  - ✓incorporates the command line options
  - ✓starts the new VM in a `virt-viewer` session

- You can see the result configuration with the command `virsh dumpxml`.

How to run a VM?

```
narges@ThinkCentreM910s:~/Downloads$ virsh list --all
 Id    Name                           State
----------------------------------------------------
 3     mykvm                          running

narges@ThinkCentreM910s:~/Downloads$ virsh undefine mykvm
Domain mykvm has been undefined

narges@ThinkCentreM910s:~/Downloads$ virsh destroy mykvm
Domain mykvm destroyed

narges@ThinkCentreM910s:~/Downloads$ virsh list --all
 Id    Name                           State
----------------------------------------------------

narges@ThinkCentreM910s:~/Downloads$ virt-install --name mykvm --memory 2048     --disk /home/narges/Downloads/ubuntu-16.04.6-desktop-i386.iso   --os-var
ERROR
An install method must be specified
(--location URL, --cdrom CD/ISO, --pxe, --import, --boot hd|cdrom|...)
narges@ThinkCentreM910s:~/Downloads$ virt-install --name mykvm --memory 2048     --disk /home/narges/Downloads/ubuntu-16.04.6-desktop-i386.iso  --import
WARNING  No operating system detected, VM performance may suffer. Specify an OS with --os-variant for optimal results.

Starting install...
```

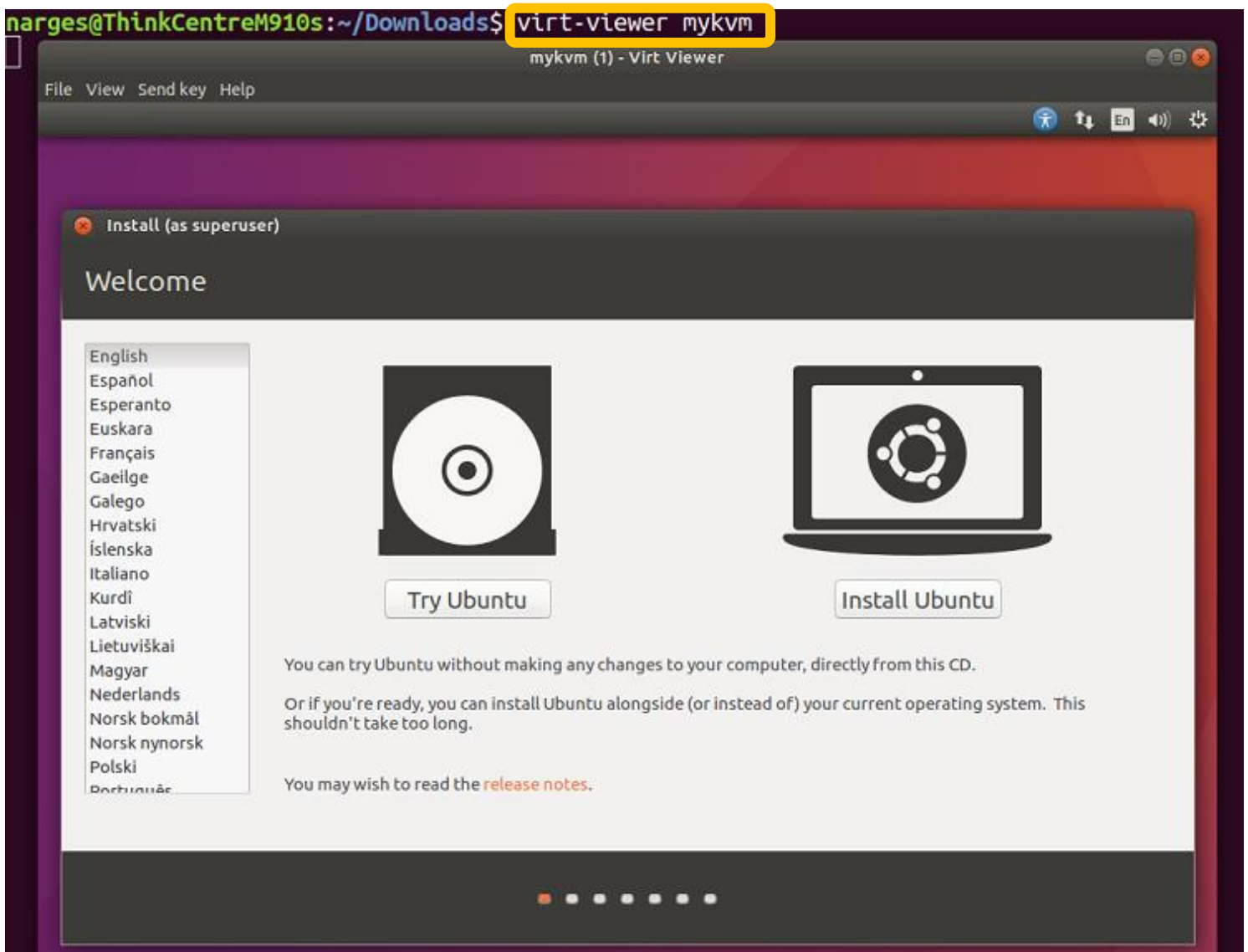


How to run a VM? (cont.)

Connect to the KVM guest to be run by the system `libvirtd` instance.

```
narges@ThinkCentreM910s:~/Downloads$ virt-install --name mykvm --memory 2048     --disk /home/narges/Downloads/ubuntu-16.04.6-desktop-i386.iso  --import
WARNING  No operating system detected, VM performance may suffer. Specify an OS with --os-variant for optimal results.

Starting install...
Domain creation completed.
You can restart your domain by running:
  virsh --connect qemu:///system start mykvm
```

Virt-viewer


narges@ThinkCentreM910s:~/Downloads$ virt-viewer mykvm
mykvm (1) - Virt Viewer
File View Send key Help
Install (as superuser)
Welcome
English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Kurdî
Latviski
Lietuviškai
Magyar
Nederlands
Norsk bokmål
Norsk nynorsk
Polski
Try Ubuntu
Install Ubuntu
You can try Ubuntu without making any changes to your computer, directly from this CD.
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.
You may wish to read the release notes.

Assignment

1. Inside your running VM, execute the lscpu and lshw commands.
   Use one of the TCP/ UDP connection protocols and return the outputs to you host and print them.

2. Bonus: write a client-server program, by socket programming (in Python or Java), between your VM and host and extend the previous assignment.

Assignment (cont.)

```
virt-install --name RHEL-6.3-LAMP \
   --os-type=linux \
   --os-variant=rhel6 \
   --cdrom
   /mnt/ISO/rhel63-server-x86_64.iso
   \
   --graphics vnc\
   --disk
   pool=NFS-01,format=raw,size=20 \
   --ram 2048 \
   --vcpus=2 \
   --network bridge=br0 \
   --hvm \
   --virt-type=kvm \
```

3. Please explain about each of the options in this `virt-install` command.

References

- https://help.ubuntu.com/community/KVM
- https://wiki.archlinux.org/index.php/KVM
- https://wiki.archlinux.org/index.php/QEMU
- https://libvirt.org/manpages/index.html
- http://old-releases.ubuntu.com/releases