ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

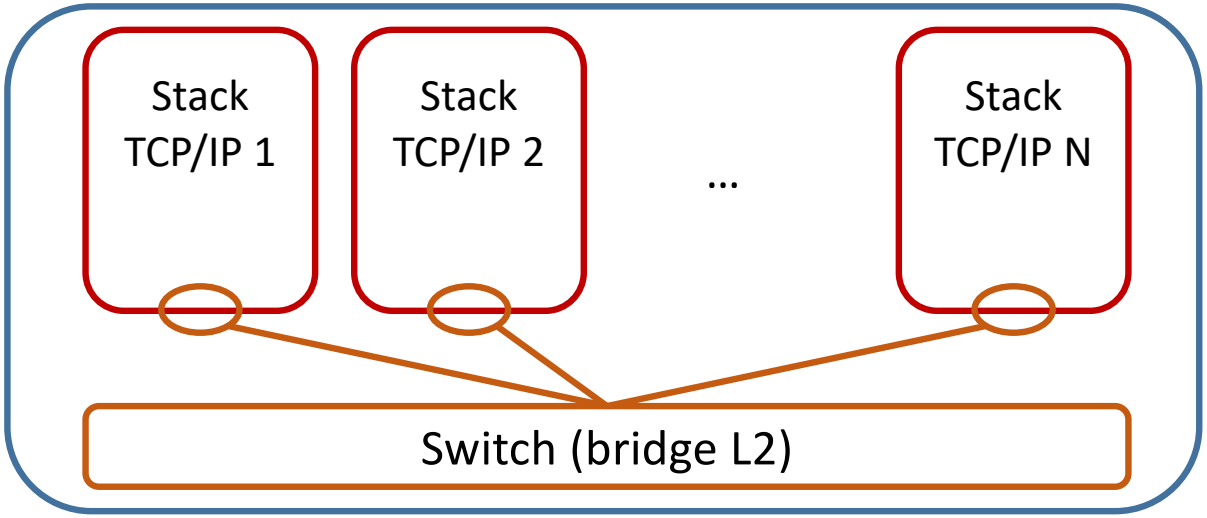
Practical experiments about IP networking and routing

Franco CALLEGATI – Walter CERRONI

AA 2019-2020

Experiments with networking

- Practical experiments in networking with physical infrastructure are complex to organize and expensive
- In the last decade:
  - Virtualization technologies in ICT gives new opportunities
- Question?
  - Can we run a whole network «inside» a single physical machine, experimenting all protocols and features «as if» in a real environment

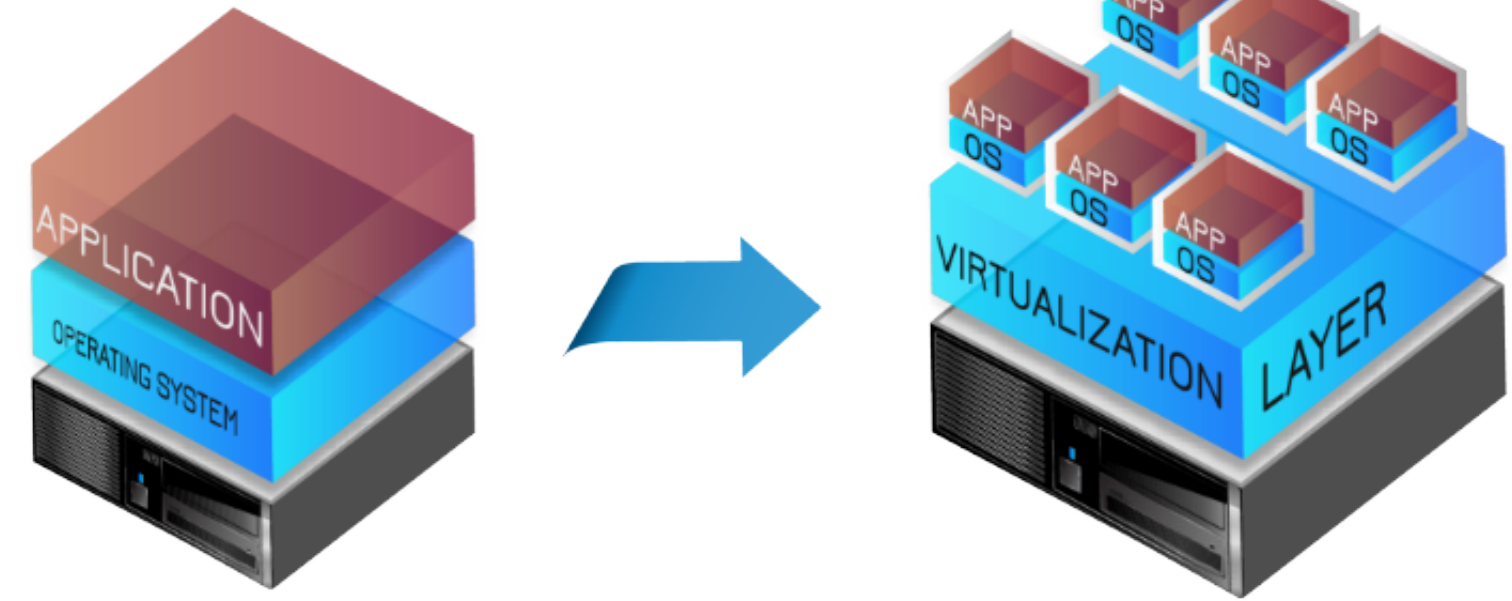


Single physical machine

Virtualization

- **Virtualization** in ICT
  - creation and use of *virtual* (i.e., abstract, software-based) versions of computing, storage and network elements
  - evolution of a concept invented in the ’60s to logically separate physical resources available on a mainframe for different applications
- Advantages
  - physical resource sharing → optimal utilization
  - decoupling software from hardware
  - scalability, flexibility, and mobility
- Issues
  - virtual resource isolation
  - security and privacy between different users

Computing element virtualization

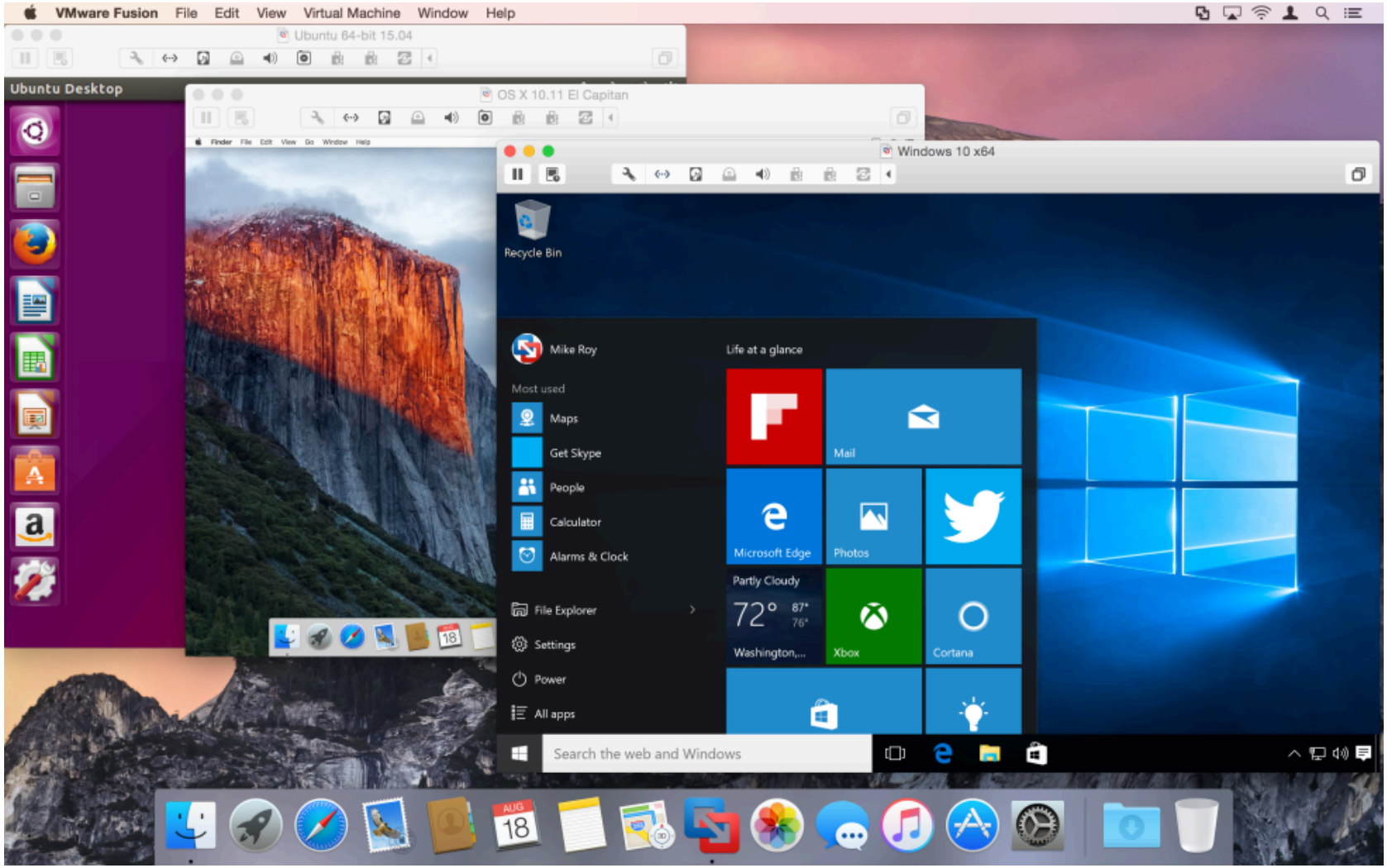


Traditional Architecture

Virtual Architecture

Source: http://exelos.com/solutions/virtualization/

Example of a virtualized system



Some dates about virtualization

- Late ’60s
  - First ideas from IBM
- 1987
  - Insignia Solutions demonstrated SoftPC, running DOS systems under Unix workstations
- 1998
  - VMWare was founded and soon started to commercialize VirtualPC
- 2007
  - Virtualbox is released

Virtualization categories

- Full virtualization
  - emulation of each HW device by the *hypervisor*
  - transparent installation of the virtual machine (VM)
  - e.g., Virtualbox, VMware, QEMU
- Paravirtualization
  - some HW functions are not emulated in SW, but shared among VMs through the hypervisor
  - VMs must be “aware” of being virtualized (driver or modified kernel)
  - e.g., Xen, KVM+Virtio
- Operating system virtualization
  - isolated “slices” of the operating system seen by applications as independent computing/storage/networking environments
  - limited VM heterogeneity (e.g., Linux on Linux)
  - e.g., Linux Containers (LXC), Docker

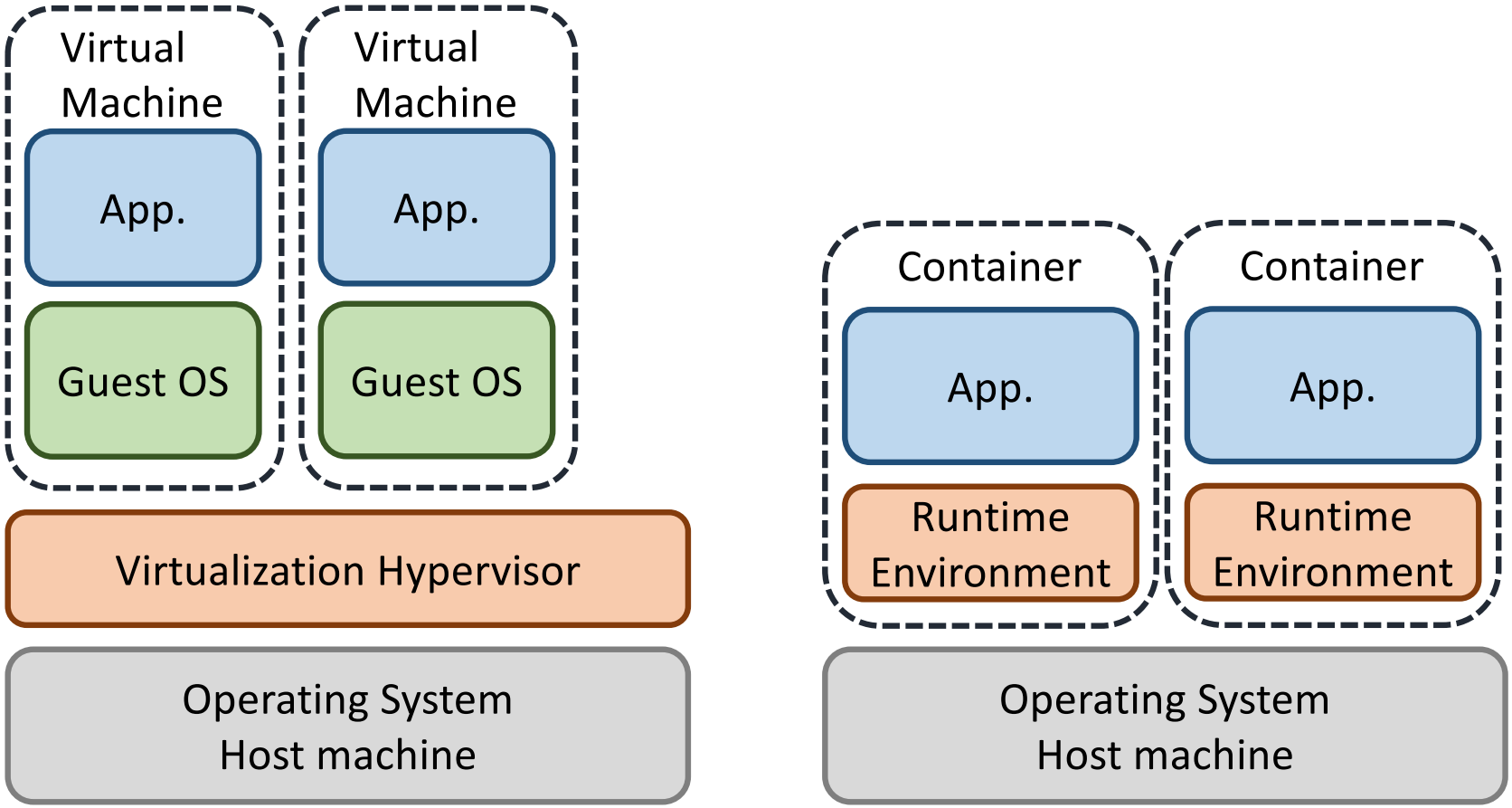
Example

- Networking with Linux namespaces
  - Run different network protocol stacks on the same machine
    - Different IP addresses
    - Different routing tables
    - Different forwarding strategies
  - Need a bridge to interconnect them

Linux containers

- Virtualization
  - Share the hardware
  - Execute different OSs on the same hardware
- Container
  - Share the hardware
  - Share the kernel of the OS
  - Execute application processes in isolation with the related environment (libraries, config files etc.)
- A VM may have a OS completely different from the host OS (linux on Windows, Windows on Linux etc.)
- A container must be compatible with the host OS

Virtualization VS Containers



Some dates about containers

- 2000
  - First ideas in FreeBSD
- 2001
  - First project on process isolation in Linux (Linux Vserver)
- 2006-2008
  - cgroups and namespaces added to Linux
- 2014
  - Linux containers (LXC)
- 2014
  - Docker

How to build LANs in the virtual space

- Networking with Linux namespaces
  - Run multiple and separate network protocol stacks on the same machine
    - Different IP addresses
    - Different routing tables
    - Different forwarding strategies
  - The IP network is built with a switch interconnecting protocol stacks by means of their interfaces
  - Need for a software switch or bridge

The key commands (1)

- `ip nets`
  - Network namespace management
    - `ip netns add NAMESPACE`
      - Create a namespace
    - `ip netns del NAMESPACE`
      - Delete a namespace
    - `ip netns list`
      - List the existing namespaces
    - `ip netns exec NAME COMMAND`
      - Execute a specific COMMAND in namespace NAME
    - `ip netns id`
      - Gives the ID of the current namespace

The key commands (2)

- `ip link`
  - Manage and configure network devices
    - `ip link add NAME type TYPE`
      - NAME is the symbolinc name of the device
      - TYPE is the type of network device
        - Veth = virtual ethernet interface
        - Bridge = Ethernet bridge (switch)
    - `ip link add NAME1 type veth peer name NAME2`
      - Create a ethernet link between two logical interfaces named NAME1 and NAME2s
    - `ip link set NAME`
      - Configure properties of NAME

The key commands (3)

- `bridge`
  - Manage bridge addressing
    - `bridge link`
      - Set properties of interfaces to links
    - `bridge fdb`
      - Manage bridge forwarding database