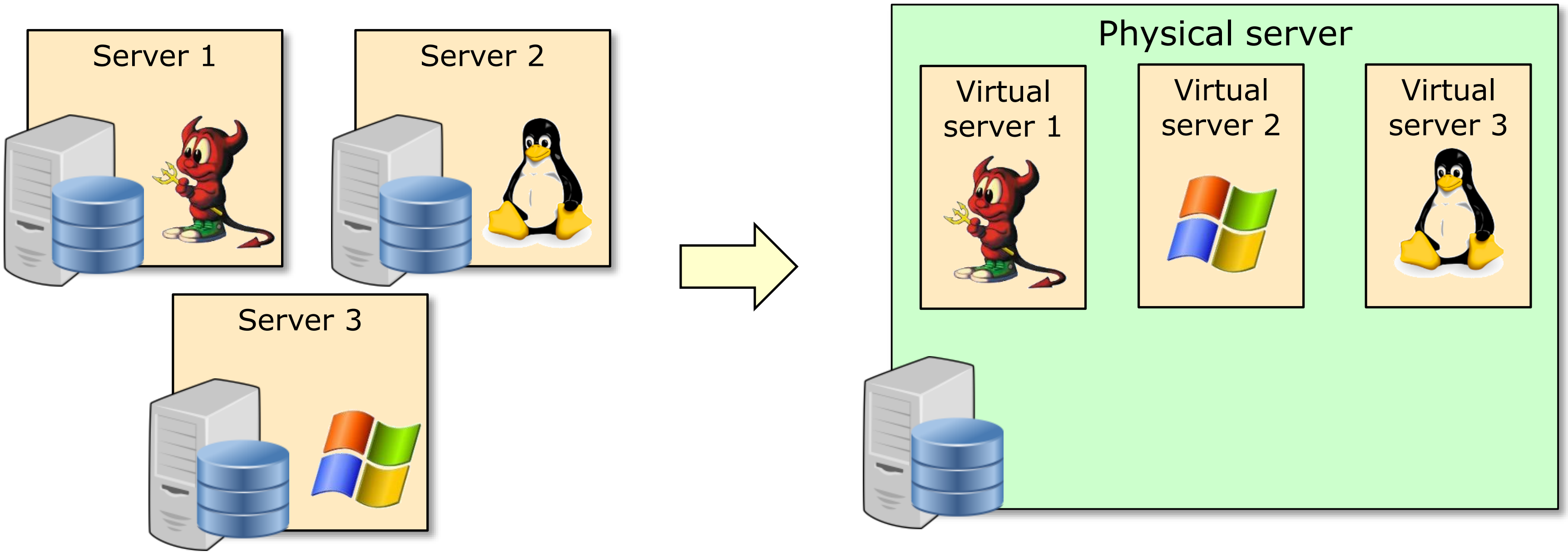
Introduction to Virtualization

Fulvio Risso

Politecnico di Torino

Introduction to computing virtualization

- A flexible way to share hardware resources between different (un-modified) operating systems
  - CPU
  - Memory
  - I/O (NICs)

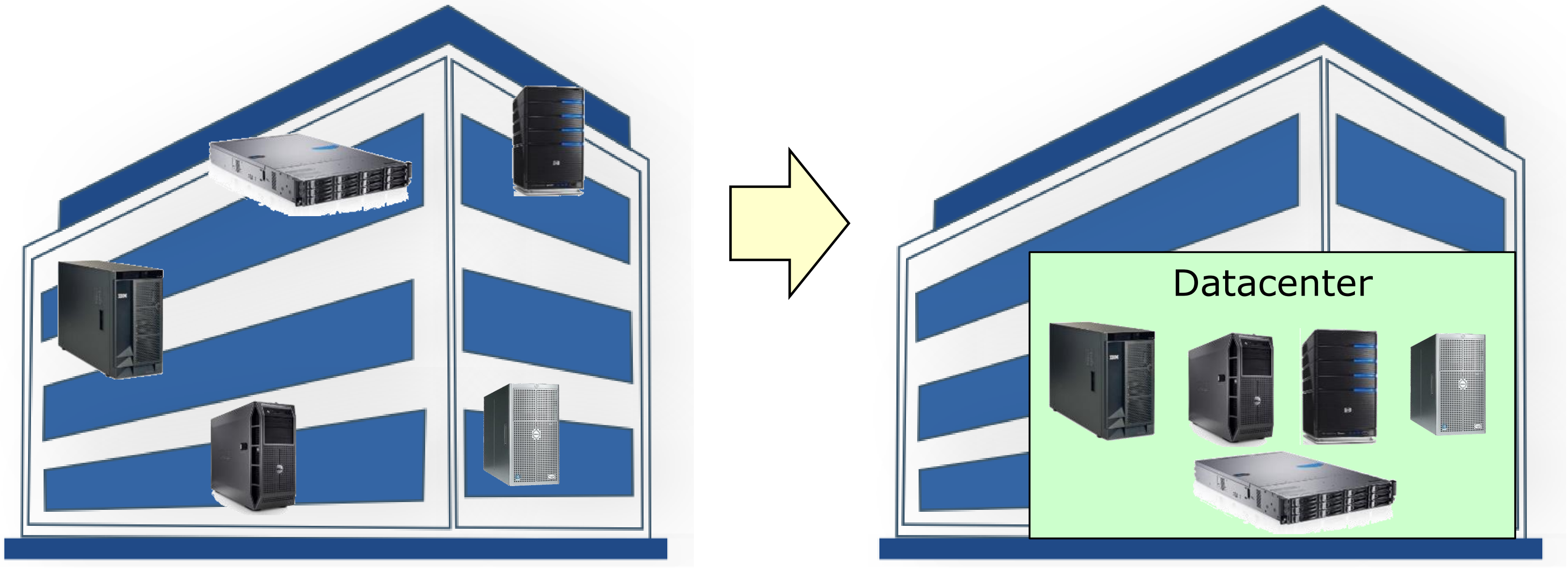


Virtualization in the old days

- Proposed for the first time around 1960 on IBM 360 to share expensive hardware resources
  - Very common in the 60’s, e.g. with mainframes
- Limited importance in 70-80's with the diffusion of cheap mini and personal computers
- Back in 90's as a new way to address new needs of reliability and consolidation of business servers
- Currently focus on x86 architecture

Virtualization in the current days (1)

- First, servers have been installed across the company
- … then, moved to datacenters in order to preserve data and make management easier…
- … and discovered that there were so many servers around…
- … and discovered that we cannot consolidate multiple apps on a few servers



Virtualization in the current days (2)

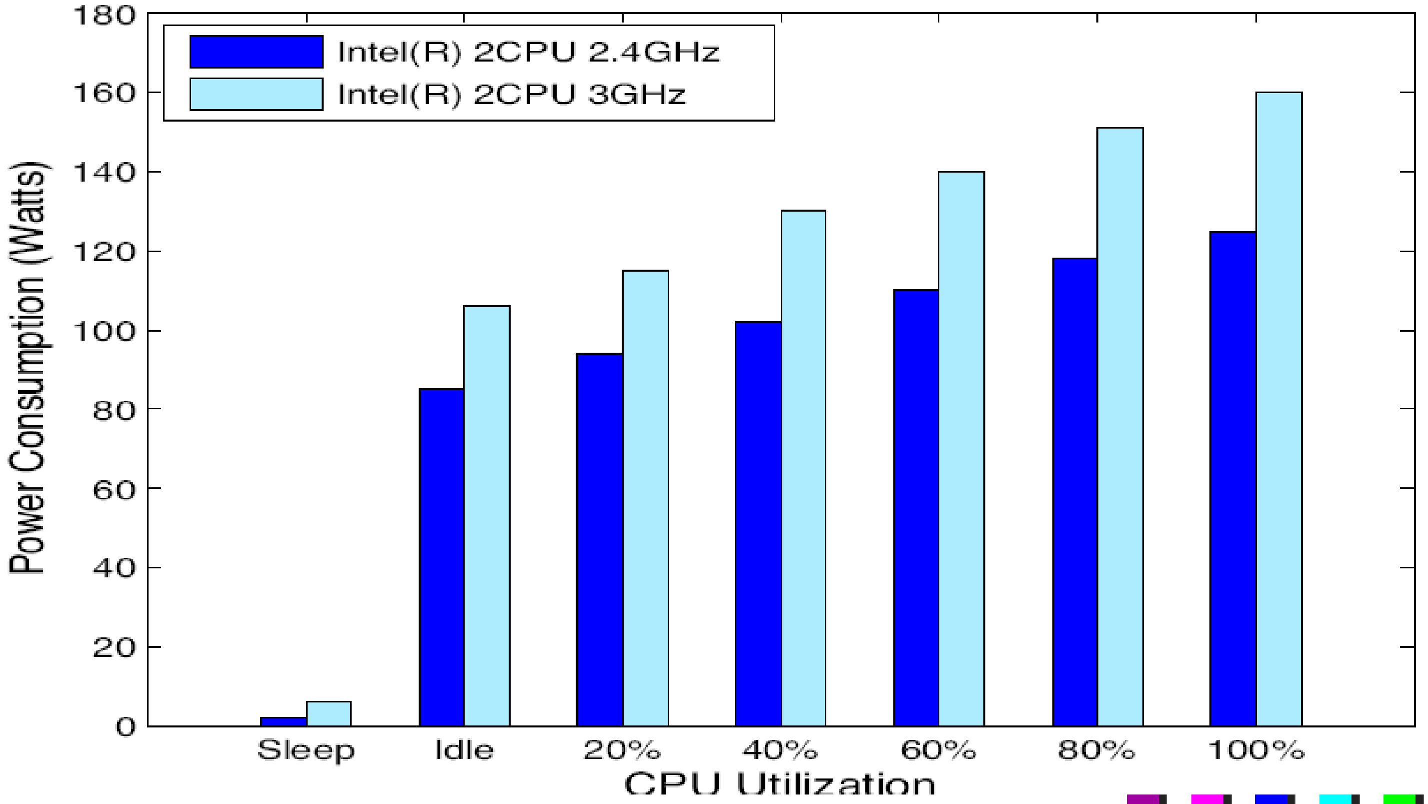
- The reason? The “one application per server” rule, due to the failure of popular OSes to provide
  - Full configuration isolation
    - E.g. A requires DLL version 1,0, B requires DLL version 2.0
  - Temporal isolation for performance predictability
    - If A is eating all the CPU, performance of B will be affected
  - Strong spatial isolation for security and reliability
    - If A crashes, it may compromise B
  - True app compatibility
    - My Application runs only on OS version X, path Y
    - This Application is certified only in this environment; if you run it in another conditions, I will not be responsible for the outcome

Virtualization in the current days (3)

- At the same time (late 90’s), the Moore Law was still valid, but… no longer with increase in the operating frequency, but with the number of cores
- Results?
- Huge amount of servers, massively underutilized, and consuming a lot of electrical power

Power consumption (1)

- Typically, 5-15% CPU utilization (per server)
- Non-linearity between CPU utilization vs power consumption
  - Better to maximize the CPU load



Power consumption (2)

- Electrical power
  - Politecnico di Torino (2005): > 3M€
- Power consumption of a single server
  - 10GbE NIC: ~15W (2008)
  - 1TB disk: ~10W idle, ~15W R/W (2008)
  - CPU: may be more than 100W
  - Memory and motherboard: ~25W
  - Idle server: ~ 66% of the peak power
    - OS still running
    - Memory, disks, motherboard, PCI slots, fans still active
  - Power supply
    - Efficiency about 75%, against 98% of the best power supplies

Power consumption and conditioning

- Typical consumption per server: ~120W
  - ~10KW per rack
  - Let’s do some math:
    - Rack footprint: 25ft^2
    - Power density: 400W/ft^2 (even more with high-end processors)
    - Typical density for commercial datacenters: 70-150W/ft^2

    → racks are only partially occupied by servers...
- Total power consumption must be multiplied by 3
  - Server power + conditioning + overheads

Server and datacenters

- Real servers are no longer important
  - Virtual servers (e.g., Server X needs 16GB RAM, …) are important, but can be created dynamically according to the actual necessities
- Consequence: we can buy tons of servers, all the same, aggregated together in a datacenter
  - Computing hardware is now a commodity

Usage scenarios

- Workstation/Desktop virtualization
  - Easy to share hardware facilities
  - Easier to restore previous configurations
  - Possibility to reproduce the same hardware/software configuration on several physical machines
- Servers virtualization
  - Several services, each one with their environment (e.g., their own OS) sharing the same hardware, with a given degree of isolation

Virtualization: features

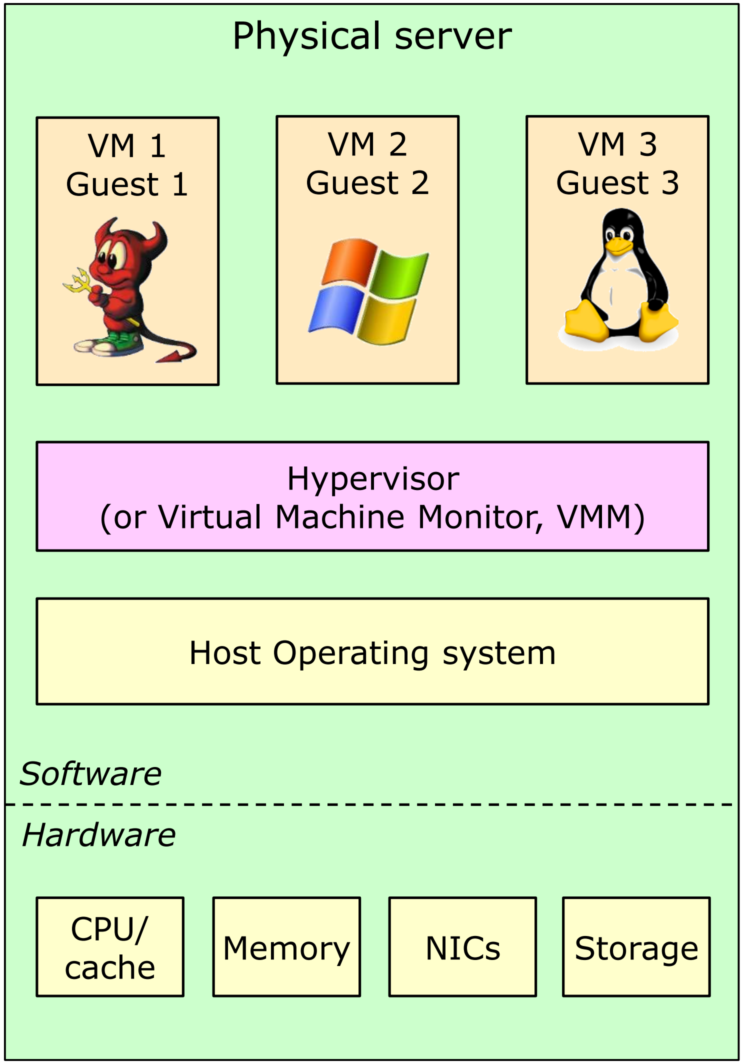
- Isolation
  - Critical applications could run in different and easily isolated OS
  - Different services could run on the same hosts with an improved degree of isolation
    - Malicious or mis-behaving (e.g., bugs) applications or services cannot compromise services running in other VMs
- Flexibility
  - Complete control over all Virtual Machine execution information
    - Possibility to pause and restart OS execution
    - Possibility to migrate the VM on another host transparently
- Consolidation
  - Completely different OSes could transparently run on the same hardware at the same time, saving hardware resources
    - Minimizing operating costs

Virtualization: limitations

- Additional overhead in running the same application
  - Each application has its own OS running, hence more requirements in terms of disks, memory and CPU
  - The additional amount of resources strongly depends on the application and technology in use
  - However, this overhead is usually considered acceptable for a wide range of different applications and operating conditions

Initial definitions

- **Virtual Machine (VM)**: software *emulation* of a physical machine that executes the software such as being in a physical one
- **Host OS**: Operating system running on the physical machine, responsible for the virtualization of the hardware platform
- **Guest OS**: Operating system running in the VM, which should not be aware of running in a virtualized environment



Hypervisor (VMM)

- The Hypervisor (or *Virtual Machine Monitor*, VMM) is the software in charge of the virtualization process
- It has to virtualize the hardware resources, such as CPU, memory, and other devices (e.g., NICs)
- Virtualization means
  - Assigning a distinct set of resources to each VM, when possible, and guaranteeing that each VM cannot get access outside its boundaries
    - E.g., the memory can be partitioned in disjoint spaces and assigned to different VMs
  - Arbitering the access to shared resources, in case those cannot be partitioned
    - E.g., unique NIC, shared among all VMs

Hypervisor (VMM)

- In practice a stripped-down OS
  - Often Linux-based
  - Native drivers manage hardware
  - A virtualization layer exports a set of “standard” devices to the upper-layer OS
    - Usually, we do not virtualize the latest video card
    - However, most important characteristics of the hardware can be exploited “natively”
    - Enable hosted OS to support a limited set of hardware
- The hypervisor may be attacked
  - Although no succedsful attacks are known right now (2008)
  - Much smaller and more defendable than a conventional OS

Types of virtualization in the x86 world

- Dynamic Binary Translation (VMware) – 1999
  - Guest OS can run unmodified in the hypervisor
- Paravirtualization (Xen) – 2003
  - Guest OS need to be modified in order to be executed
- Hybrid Virtualization (KVM, Hyper-V, VMware Esxi, Xen HVM) – 2006
  - Hypervisors exploit newly introduced primitives in the host OS, which allows them to be executed inside an existing operating system

Virtualization benefits

- Server consolidation
  - Exploit multi-core CPUs
  - Optimize energy consumption
    - Huge savings
    - Green calculator: http://www.vmware.com/solutions/green/calculator.html
- Decoupling physical hardware from logical servers
  - Rapid deployment of new servers
  - Move servers between different hardware
    - Either as image, or with server running (e.g., VMware VMotion)
  - Capability to give more CPU cycles to servers that require more power
    - Dynamic load balancing between server
  - Disaster recovery
    - Either static (move images) or dynamic (replace faulty server)
- Management
  - Secure remote console
  - Reboot / power control
  - Performance monitoring
  - Easier to setup a new server (no need to deal with different physical hardware)

From server virtualization to cloud computing

- VMs enable a much more flexible use of the resources
- Servers are being consolidated in data centers
  - Standard servers, instead of many tipes of similar machines
- Public datacenters appeared
- Cloud computing was born

Cloud Computing

- Started in 2006 at Amazon
  - The “data-centers” revolution
- Generate income form unused resources
- Elastic and on-demand provisioning of computing (and storage/networking)
- Key advantages:
  - Ease of scaling up/down
  - Consolidation
  - Availability
- Mostly based on computing virtualization

Cloud Computing taxonomy

- Use model
  - Infrastructure as a Service
  - Platform as a Service
  - Software as a Service
- Deployment model
  - Public
  - Hybrid
  - Private

Managing cloud computing resources

- VMs, flexible and secure modules but…
  - What if we have tons of servers around, and we would like to coordinate the placement of VMs across the entire infrastructure?
  - How can we manage a large server infrastructure, e.g., starting / stopping / relocating VMs, without having to deal with each single server?
  - How can we migrate VMs from one server to another?
  - How can we split the infrastructure (storage, compute, network) among multiple tenants?
- We need a software framework that allow the data center administrator to control the entire infrastructure
  - “Cloud Toolkit”: unified interface to control the data center

Cloud toolkit

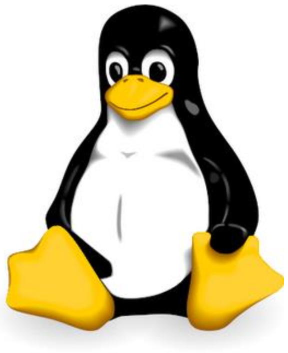
- Provides an higher-level abstraction of resources
- Communicates with different and heterogeneous technologies
  - Hypervisor (Compute)
  - Storage systems
  - Network resources (Controller SDN, virtual switches…)
- Exports North-bound APIs to user applications to scale up/down
- Provides the operating system for cloud computing virtualization infrastructures

Cloud toolkit examples

- Several open-source
  - Eucalyptus
    - Complete AWS-EC2 compatibility
  - OpenNebula
    - EU funded research project
  - Apache CloudStack
    - Formerly Cloud.com software stack
  - OpenStack
    - Rackspace and Nasa, 2010
    - Ubuntu, 2011
    - RedHat, 2012
    - Many others joining forces
- Proprietary technologies
  - VMware vSphere (formerly VMware Infrastructure 4)

Linux vs. OpenStack

- 1990-2010
  - Software is created to be executed on a single host
- Linus is the de-facto open-source operating system for a single host



- 2010 - ?
  - Software is being created to be executed on the cloud
  - Existing software is increasingly ported on the cloud (e.g., through VMs)
- OpenStack could become the de-facto open-source operating system for the cloud