

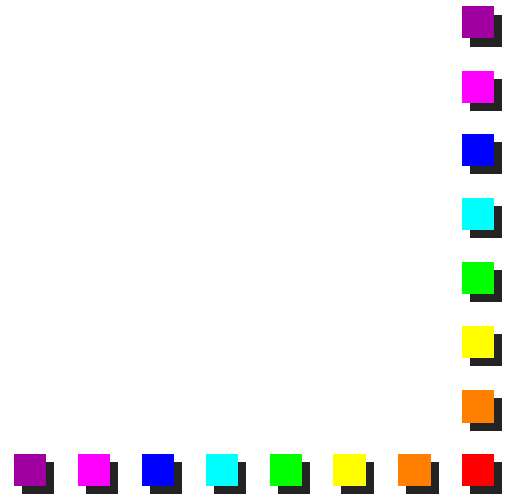


Introduction to Computing Virtualization

Fulvio Riso

Politecnico di Torino

<https://fulvio.frisso.net>





Copyright notice

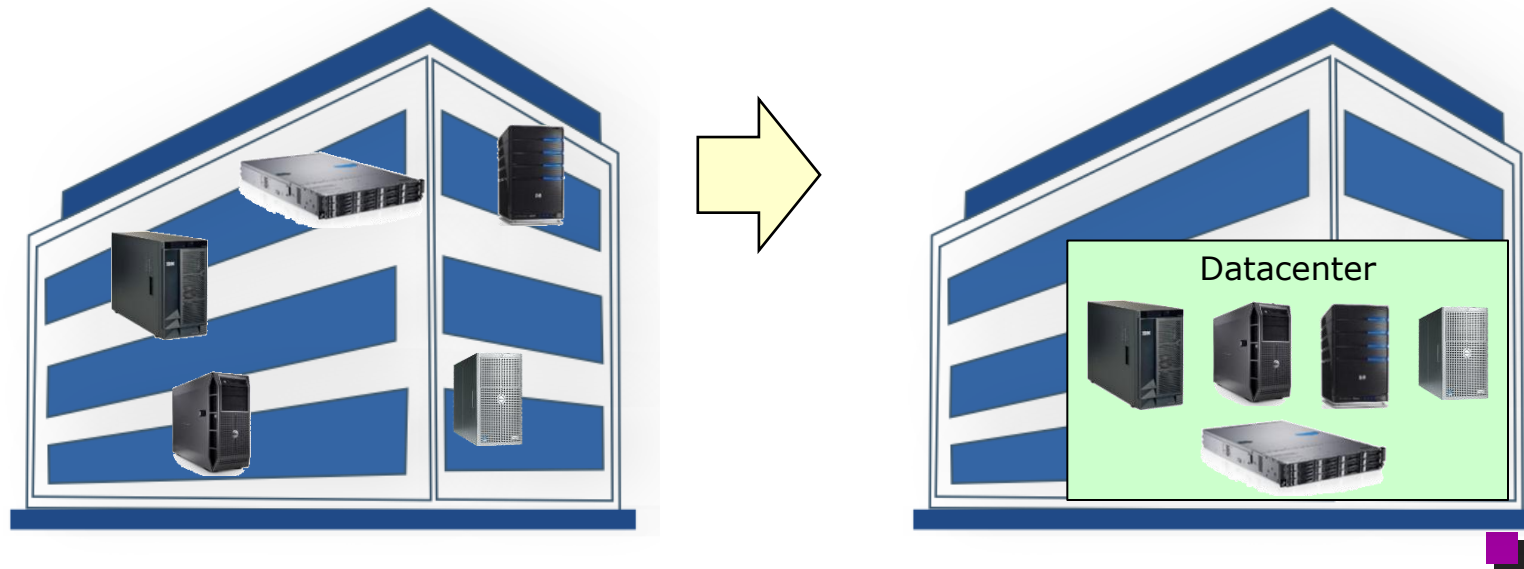
- This set of transparencies, hereinafter referred to as slides, is protected by copyright laws and provisions of International Treaties. The title and copyright regarding the slides (including, but not limited to, each and every image, photography, animation, video, audio, music and text) are property of the authors specified on page 1.
- The slides may be reproduced and used freely by research institutes, schools and Universities for non-profit, institutional purposes. In such cases, no authorization is requested.
- Any total or partial use or reproduction (including, but not limited to, reproduction on magnetic media, computer networks, and printed reproduction) is forbidden, unless explicitly authorized by the authors by means of written license.
- Information included in these slides is deemed as accurate at the date of publication. Such information is supplied for merely educational purposes and may not be used in designing systems, products, networks, etc. In any case, these slides are subject to changes without any previous notice. The authors do not assume any responsibility for the contents of these slides (including, but not limited to, accuracy, completeness, enforceability, updated-ness of information hereinafter provided).
- In any case, accordance with information hereinafter included must not be declared.
- In any case, this copyright notice must never be removed and must be reported even in partial uses.

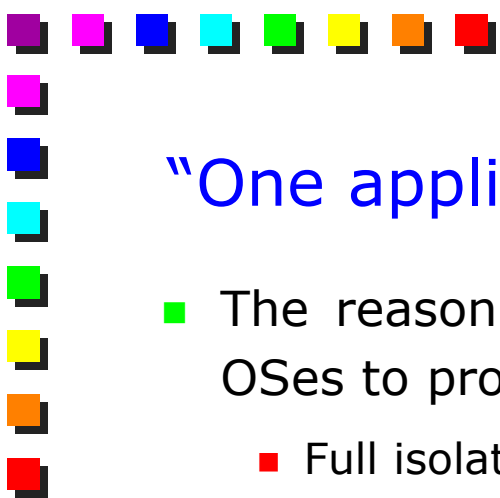




The path toward computing virtualization

- Initially (~1980-1990), servers were installed “randomly” in the company buildings
- ... then, moved to datacenters in order to preserve data and make management easier...
- ... and discovered that there were so many servers around, mostly idle...
- ... and that we cannot consolidate multiple apps on a single server (see later)
- ... and that servers have so many unused CPU cores (see later)





“One application per server” rule

- The reason? The “one application per server” rule, due to the failure of popular OSes to provide:
 - Full isolation of configuration/shared components
 - App A requires shared library v. 1.0, App B requires v. 2.0
 - A is certified by the vendor only on OS version X, patch Y
 - If A runs in a different environment, the vendor will not be responsible of possible misbehavior
 - Temporal isolation for performance predictability
 - If A consumes a lot of CPU, performance of B will be affected
 - If A sends a lot of traffic, performance of B will be affected
 - Strong spatial isolation for security and reliability
 - If A crashes, it may compromise B
 - A can send traffic directly to B, bypassing the (outside) firewall

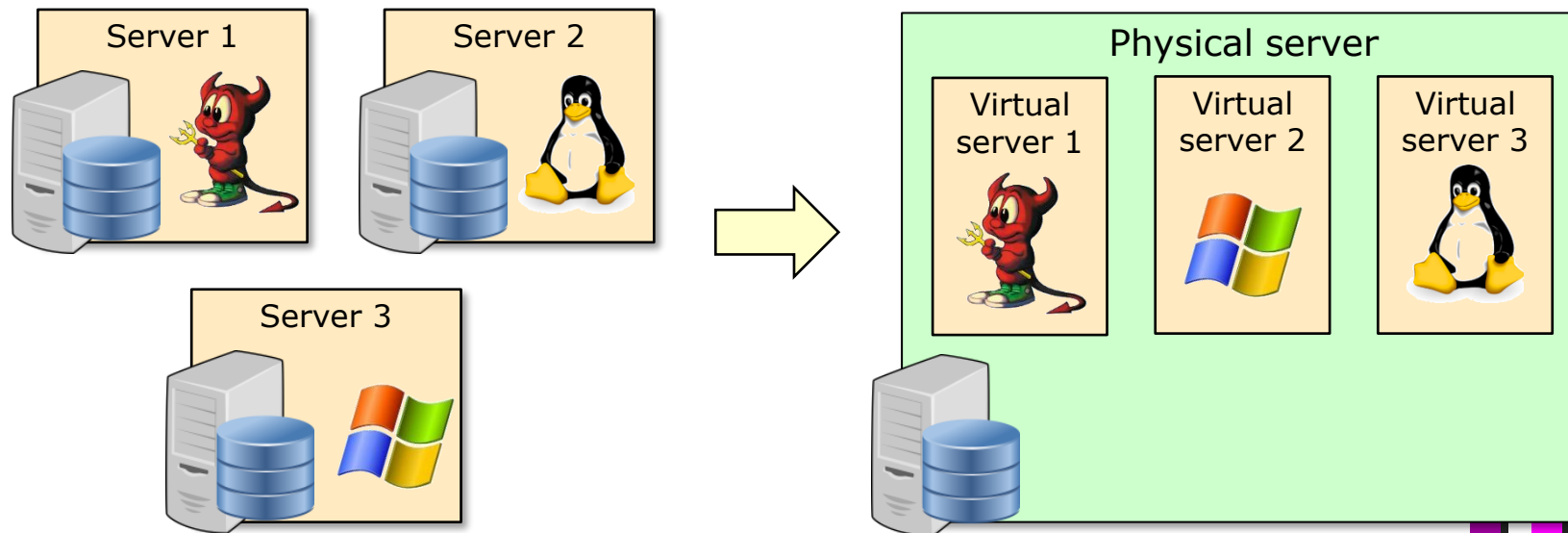
Multi-core servers, vs single (-threaded) apps

- Impossibility to consolidate multiple apps on the same server was a problem also for another reason
 - At the same time (late 90's), the Moore Law was still valid, but... no longer with the increase of the operating frequency (clock), but with the number of CPU cores
 - Results? Huge amount of servers, massively underutilized, and consuming a lot of electrical power
 - A lot of CPU cores available, but not used because of the "one app per server" rule
 - A lot of apps running as single-threaded, hence not exploiting the CPU resources available



Computing virtualization in a nutshell

- Computing Virtualization is a flexible way to share hardware resources between different (un-modified) operating systems
 - CPU
 - Memory
 - I/O (NICs + Disks)
- This is also called “**lift and shift**” approach: “lift” (separate) software from hardware, and “shift” it (unmodified) to virtual servers





Virtualization in the past

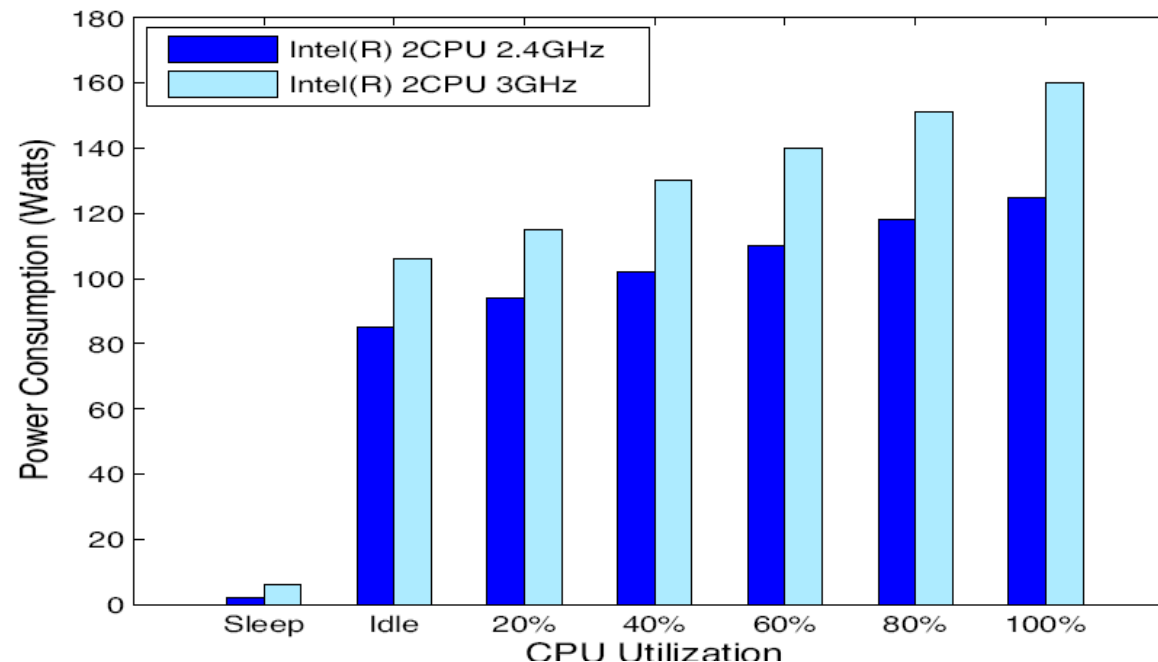
- Proposed for the first time around 1960 on IBM 360 to share expensive hardware resources
 - Very common in the 60's, e.g. with mainframes
- Limited importance in 70-80's with the diffusion of cheap mini and personal computers
- Back in 90's as a new way to address new needs of reliability and consolidation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





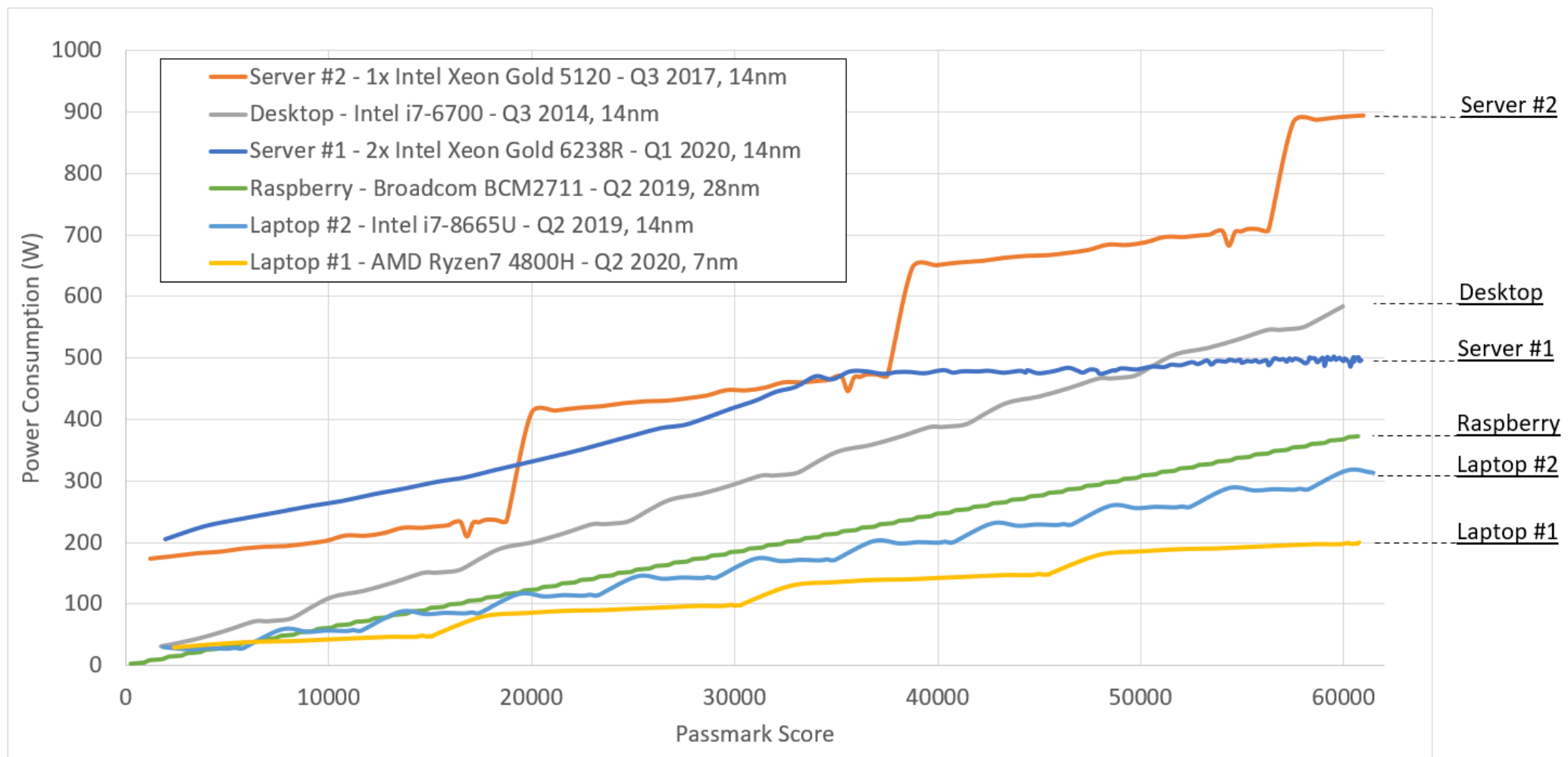
Power consumption in servers: a single-server view

- Non-linearity between CPU utilization vs power consumption
 - Power suppliers, memory, chipset, NICs, etc. consume energy independently from the CPU load
- One big step when powering on the server
- (After), linear increase on the CPU load





Power consumption: comparing different machines





Power consumption : considerations

- Usually, 5-15% CPU utilization (per server)
- Better to use less servers, and maximize the CPU load on them
 - Up to some extent, to avoid worsening the QoE of our users: an excessive load can bring to resource contention (e.g., disk, memory, CPU cache, etc)
- Important differences between different machines in terms of energy efficiency against the same workload
 - E.g., laptops look more efficient, but we pay in other directions (e.g., less robust to high, constant workloads)
 - E.g., ARM cores look more efficient than x64, but the absolute performance per core is usually smaller (e.g., smaller clock speed)



Power consumption: some numbers

- 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: ~100-300W
 - Memory and motherboard: ~25W
 - Idle server: could be 50% of the peak power
 - OS still running
 - Memory, disks, motherboard, PCI slots, fans still active
 - Power supply: could reach more than 100W (often x2) particularly if energy-hungry cards are present (e.g., GPU for machine learning, object recognition, etc., which require power supplies of ~1-1.5KW)
 - Efficiency about 75%, against 98% of the best power supplies



Power consumption and conditioning

- Typical consumption per server: $\sim 120\text{W}$ (and more)
 - $\sim 10\text{KW}$ per rack
 - Let's do some math:
 - Rack footprint: 25ft^2
 - Power density: $400\text{W}/\text{ft}^2$ (even more with high-end processors)
 - Typical density for commercial datacenters: $70\text{-}150\text{W}/\text{ft}^2$
 - $\rightarrow$ racks are only partially occupied by servers...
- In the old days, total power consumption must be multiplied by 3
 - Server power + conditioning + overheads
- Now, a "normal" datacenter can be $\sim \times 1.8$
- An hyperscaler can reach $\sim \times 1.2$
- PUE (Power Usage Effectiveness)

(data 2008)

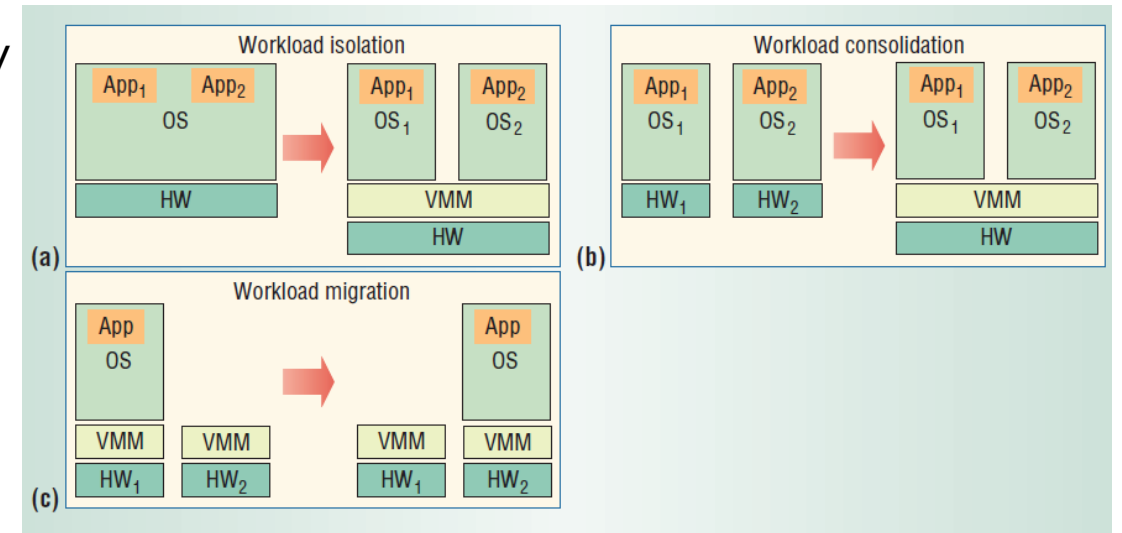
Virtualization: advantages (1)

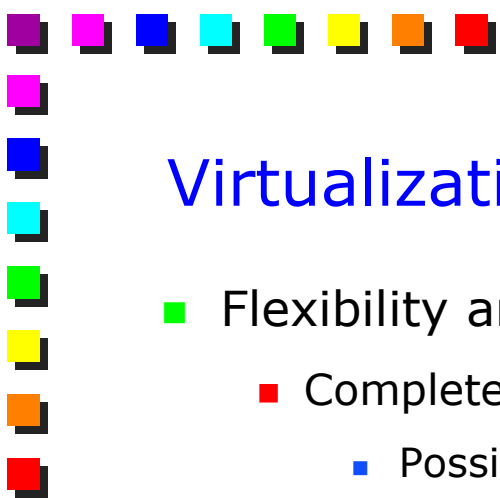
■ Isolation

- 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 misbehaving (e.g., bugs) applications or services cannot compromise services running in other VMs
 - We can assign different CPU cores to different VMs

■ Consolidation

- Completely different OSes could transparently run on the same hardware at the same time, saving hardware resources
 - Minimizing operating costs
- Exploit multi-core CPUs
- Optimize energy consumption





Virtualization: advantages (2)

■ Flexibility and Agility

- Complete control over the execution information of all VMs
 - Possibility to pause and restart OS execution
 - Possibility to migrate the VM (virtual server) to another host (even when running), e.g., to consolidate servers (e.g., during night), or to give it more resources (when a physical machine becomes overloaded)
 - Possibility to duplicate a running VM, e.g., to address a peak load
 - Disaster recovery
 - Rapid deployment of new servers
 - Spawning new instances
- 



A few more works about “Agility”

- According to some people (e.g., the author of these slides), agility is the most important advantage at all
 - Instead of playing with the real world (servers, cables, etc.) you can create your own world dynamically, with a few lines of codes (or mouse clicks)
- This enables the creation of “autonomic” systems, which autonomously react to a set of stimulus, and dynamically evolve and adapt to the changing environment
 - Clever response can be enabled by Artificial Intelligence and such
 - However, the capability to evolve is provided by the infrastructure underneath, thanks to this agility property

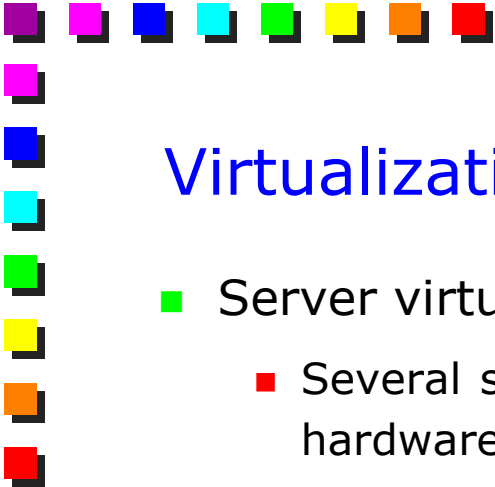




Virtualization: limitations

- 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
- More difficult to handle heterogeneous hardware
 - E.g., offering the access to special components (e.g., powerful GPUs, hardware offloading cards, etc.) to some applications





Virtualization: usage scenarios

- Server virtualization
 - Several services, each one with their environment (e.g., their own OS) sharing the same hardware, with a given degree of isolation
- 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



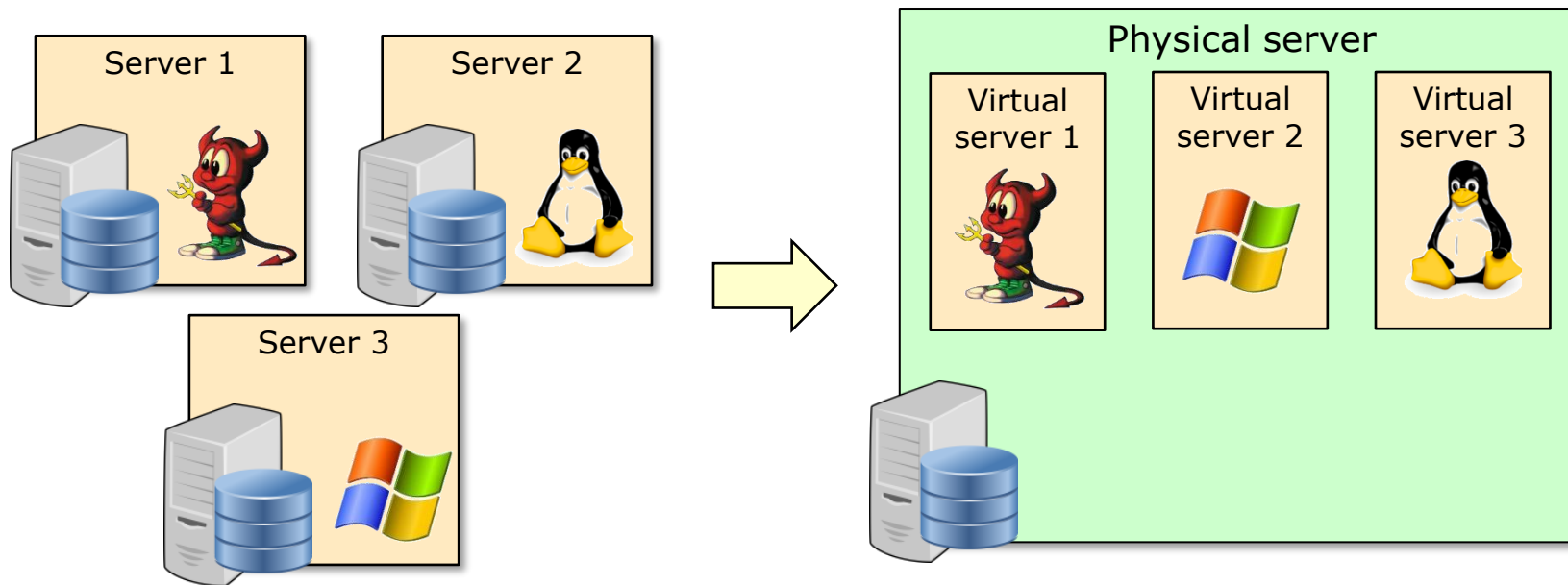
Virtualization: economic considerations

- Server virtualization is definitely more important (from economic viewpoint)
- Workstation/Desktop virtualization is not an important economic driver; in fact, most of the products in that area are free
- Main products:
 - Server virtualization: Vmware ESXi, Microsoft Hyper-V, Citrix XenServer, Linux KVM
 - Workstation/Desktop virtualization: Oracle Virtualbox, Linux KVM, Vmware Workstation



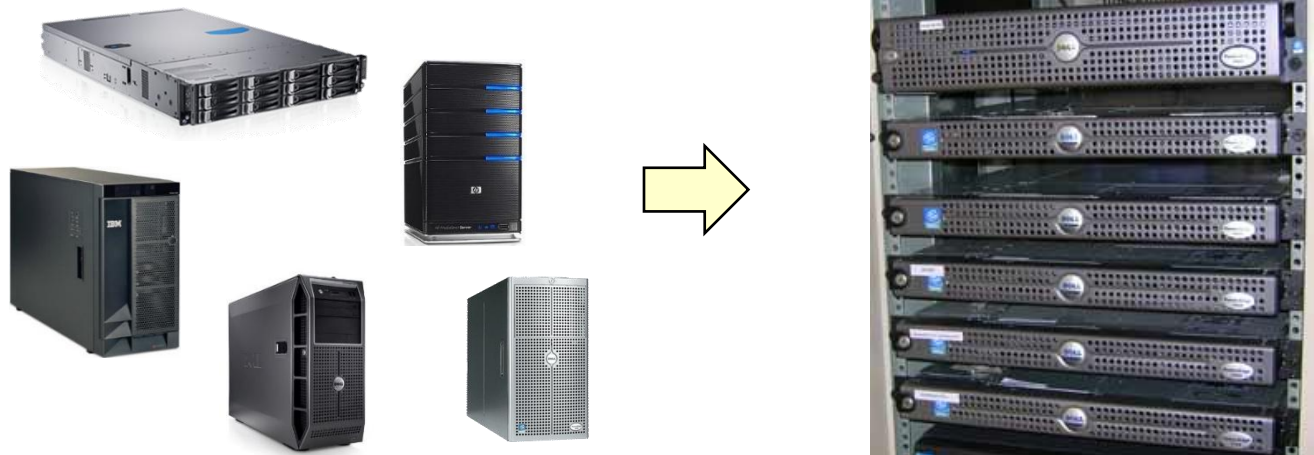
Common Off The Shelf (COTS) hardware (1)

- Focus has now shifted from *real* servers to *virtual* servers
 - No longer important to have a physical machine with a given set of characteristics (e.g., powerful CPU, 16GB RAM, etc.)
 - We can create dynamically a virtual server with the requested characteristics (e.g., CPU, memory, disks, number of NICs, ...), based on the actual necessities



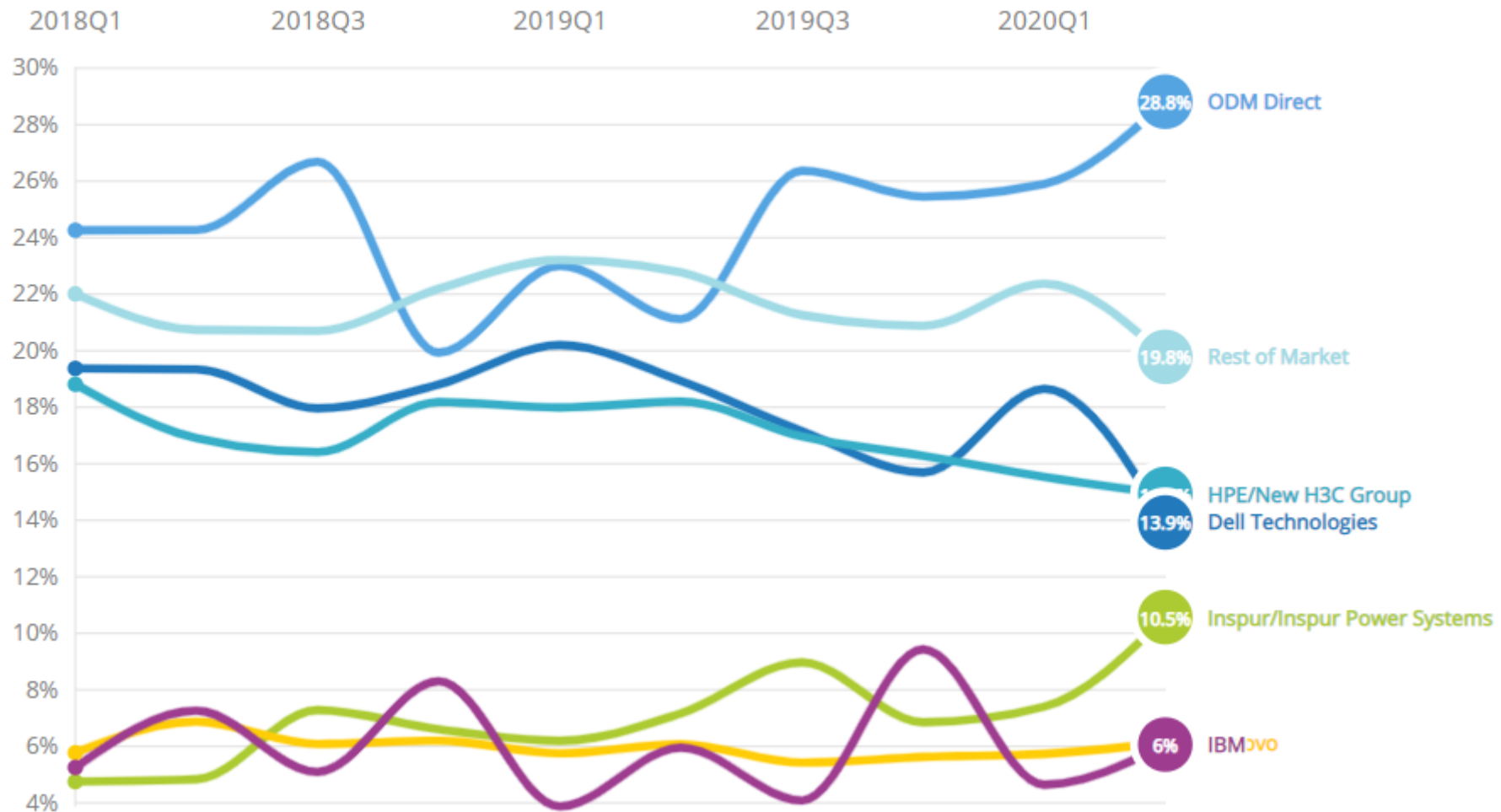
Common Off The Shelf (COTS) hardware (2)

- Consequence: (corporate) users can buy tons of equivalent servers, with exactly the same hardware characteristics, and:
 - Aggregate them together in a datacenter
 - Virtualize their resources in order to create the specific set of virtual servers (a.k.a., VMs) we need
- Computing hardware becomes a commodity
 - Just buy the “cheapest” (e.g., *white label*) COTS hardware, asking for volume discounts



Server market share (1)

Worldwide Top 5 Server Companies, 2018Q1 - 2020Q2 Vendor Revenue Market Share



<https://www.idc.com/promo/servers> [update: Sept 2020]

OEM vs ODM

OEM (Original Equipment Manufacturing)

The product is fully designed by one company and then licensed out to a manufacturer to produce.

The Apple iPhone, for example, was invented and designed by Apple, and then licensed out to Foxconn to produce. As a result, the iPhone enjoys a higher level of product differentiation because its design is only available to Apple and its licensed manufacturer.

ODM (Original Design Manufacturing)

The product is totally engineered by a third party, which may sell it under different brands or even through its own name.

An importer can select an already-existing product design from a factory catalog, makes a few small changes and sells it under their own brand name. Changes can include things like packaging or product bundles, colors and branding, and some limited adjustments to components or functionality.

It is also known as "private labeling."

Some vendors (servers and network equipment):

- Quanta: <https://www.qct.io/product/overview/Server>
- Fiberstore: <https://www.fs.com/c/networking-911>



Partially adapted from:
<https://www.guidedimports.com/blog/odm-oem/>

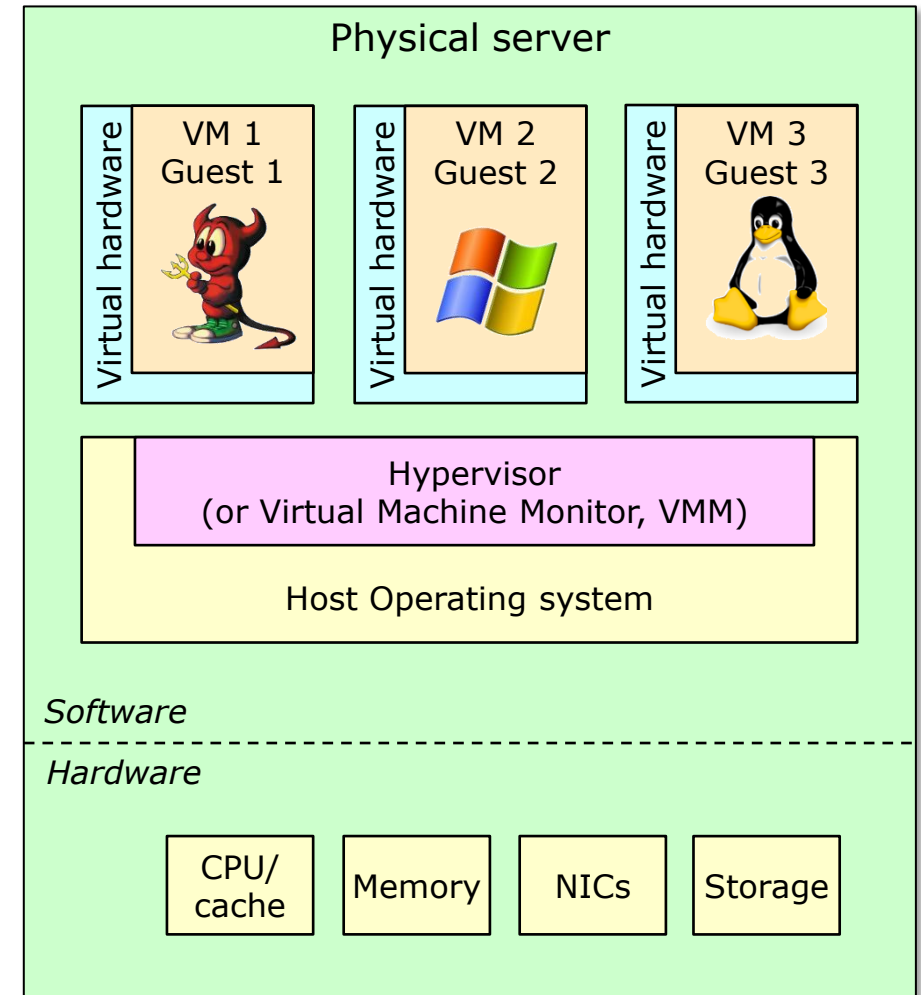


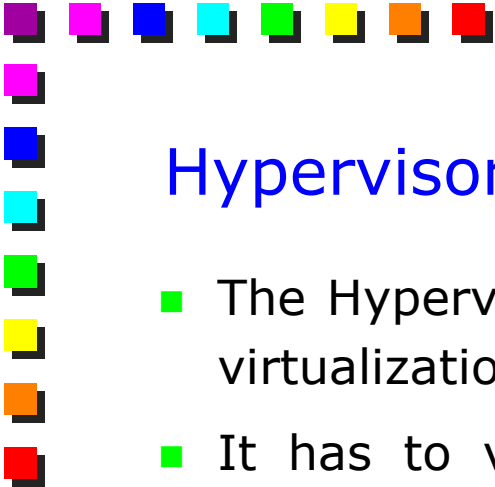
ODM Servers ("white box"): manufacturers, buyers

- Mostly used by big companies
 - Interesting analysis from IDC, "Worldwide 2019 Enterprise ODM Direct Infrastructure Market Value Surpasses \$21 Billion, According to IDC," Jun 2020
 - <https://www.idc.com/getdoc.jsp?containerId=prUS46594420>
 - "The leading ODM Direct vendors (when measured and ranked by the value of systems shipped) were: Quanta, Wiyynn, Foxconn, Inventec, MiTAC, ASRock Rack, and Gigabyte. These seven vendors accounted for 89% of the \$21.2 billion in 2019 sales."
 - Top 7 Cloud Providers: Amazon, Microsoft, Facebook, Google, Alibaba, Tencent, Baidu.
 - Interesting article from Network World: "Is there a white-box server in your data center's future?", Oct 2018
 - <https://www.networkworld.com/article/3311839/is-there-a-white-box-server-in-your-data-center-s-future.html>
 - "Sales from those non-big-three vendors [i.e., Dell, HPE, Lenovo] are going almost exclusively to hyperscale data-center vendors like Amazon, Google, Microsoft, Facebook, Equinix, Cloudflare and other companies building data centers measured in acres."
- 

Initial definitions

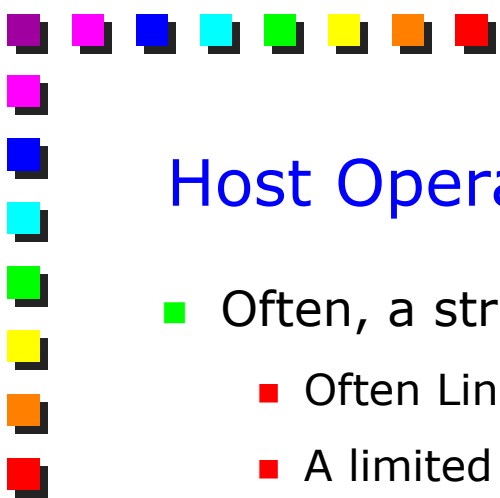
- Virtual Machine (VM)
 - Software emulation of a physical machine that executes OS + Apps such as being in a physical one
- Host OS
 - OS running on the physical machine, in charge of the virtualization of the hardware
 - It includes the Hypervisor (next slide)
- Guest OS
 - OS running in the VM, which should not be aware of running in a virtualized environment





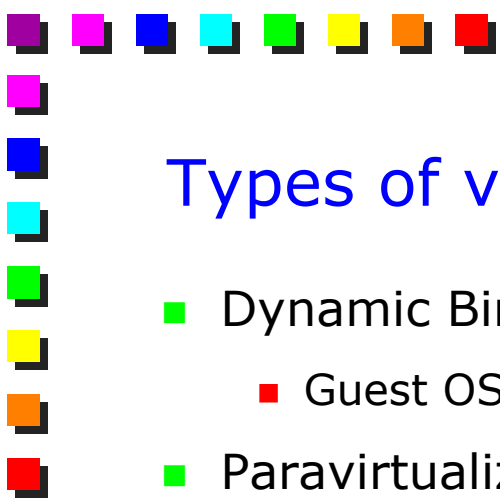
Hypervisor (VMM)

- 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
 - Arbitrating the access to shared resources, in case those cannot be partitioned
 - E.g., unique NIC, shared among all VMs
- 



Host Operating System

- Often, a stripped-down OS
 - Often Linux-based
 - A limited set of native drivers to manage the hardware
 - The virtualization layer (VMM/Hypervisor) 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 Host OS may be attacked
 - Given its characteristics, it is usually much smaller and more defensible than a conventional OS
- 

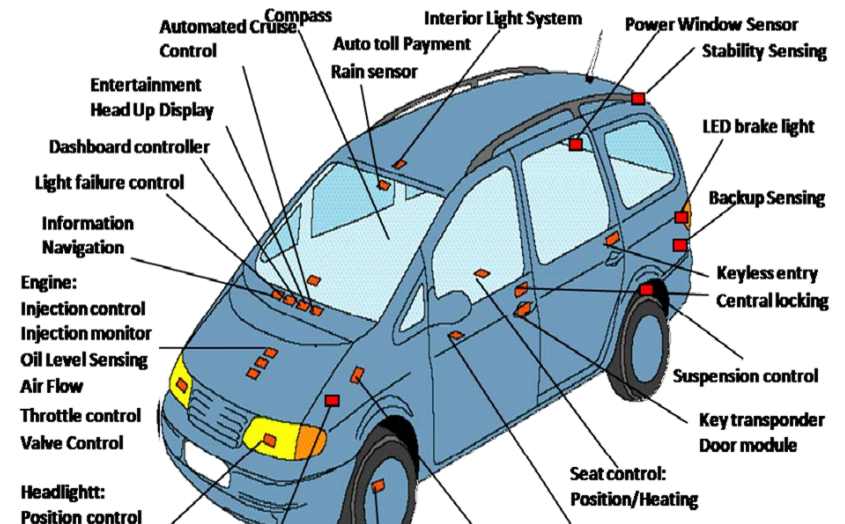


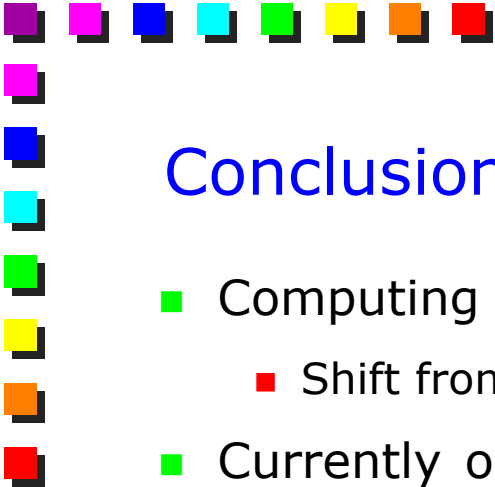
Types of virtualization in the x86 world

- 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 (all) – 2006
 - Hypervisors exploit newly introduced primitives in the host OS, which allows them to be executed inside an existing operating system
 - In addition, they exploit hardware features now included in the CPU/chipset

Is virtualization only for servers?

- Economic traction privileges server virtualization
 - Desktop virtualization is less important
- What about the embedded world?
 - As far as the problems are the same (application isolation, compatibility with different OS, etc), virtualization can provide an excellent solutions
 - Newer embedded-oriented CPUs already integrate hardware support for virtualization
 - Possible use case: decrease the amount of embedded controllers/CPUs in a car (currently ~100)





Conclusions

- Computing virtualization is a very important technology nowadays
 - Shift from hardware servers to software servers, e.g., from *how* to do, to *what* to do
- Currently ongoing shift toward more lightweight virtualization technologies (a.k.a. containers)
- Another problem arises: the necessity for the cooperation of many (distant?) physical servers
 - This will be *cloud computing*