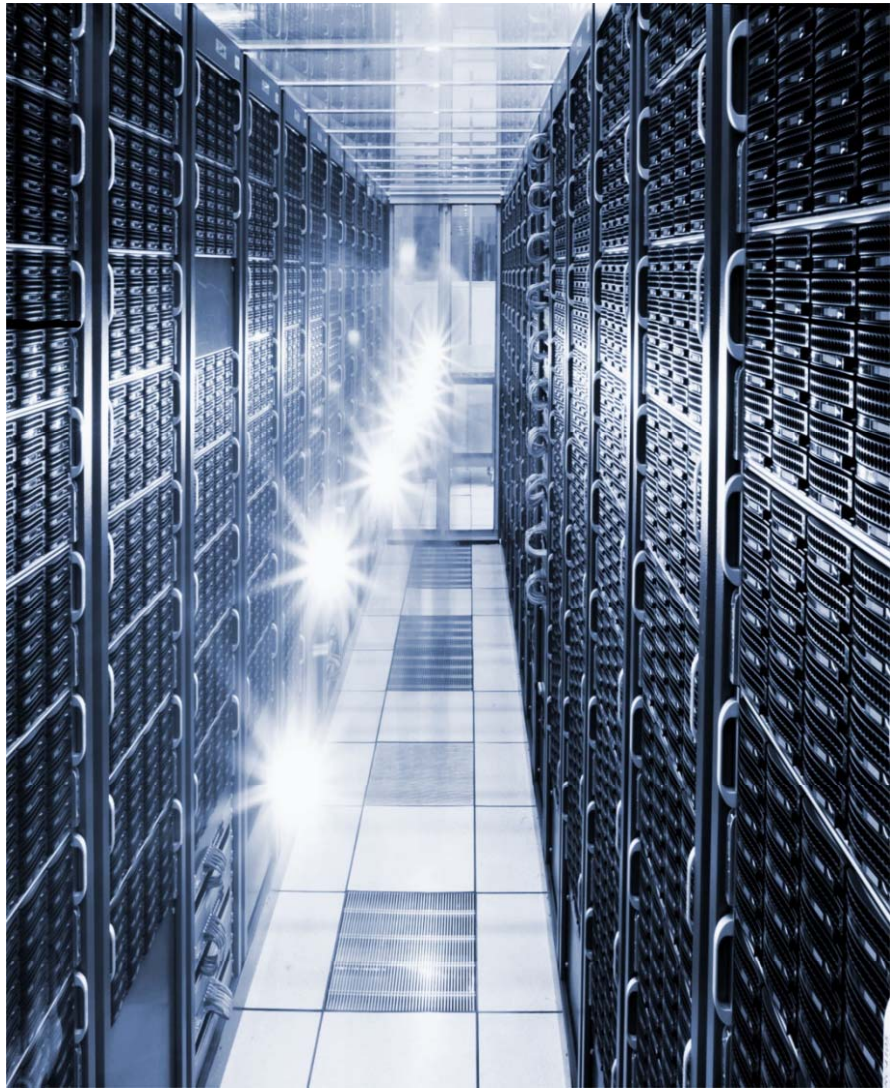
Virtualization

Carlos Garcia Fernandez, CERN

Luigi Gallerani, CERN

Inverted CERN School of Computing, 3-4 March 2011

Virtualization



Carlos Garcia Fernandez

Luigi Gallerani

carlos.garcia.fernandez@cern.ch
luigi.gallerani@cern.ch

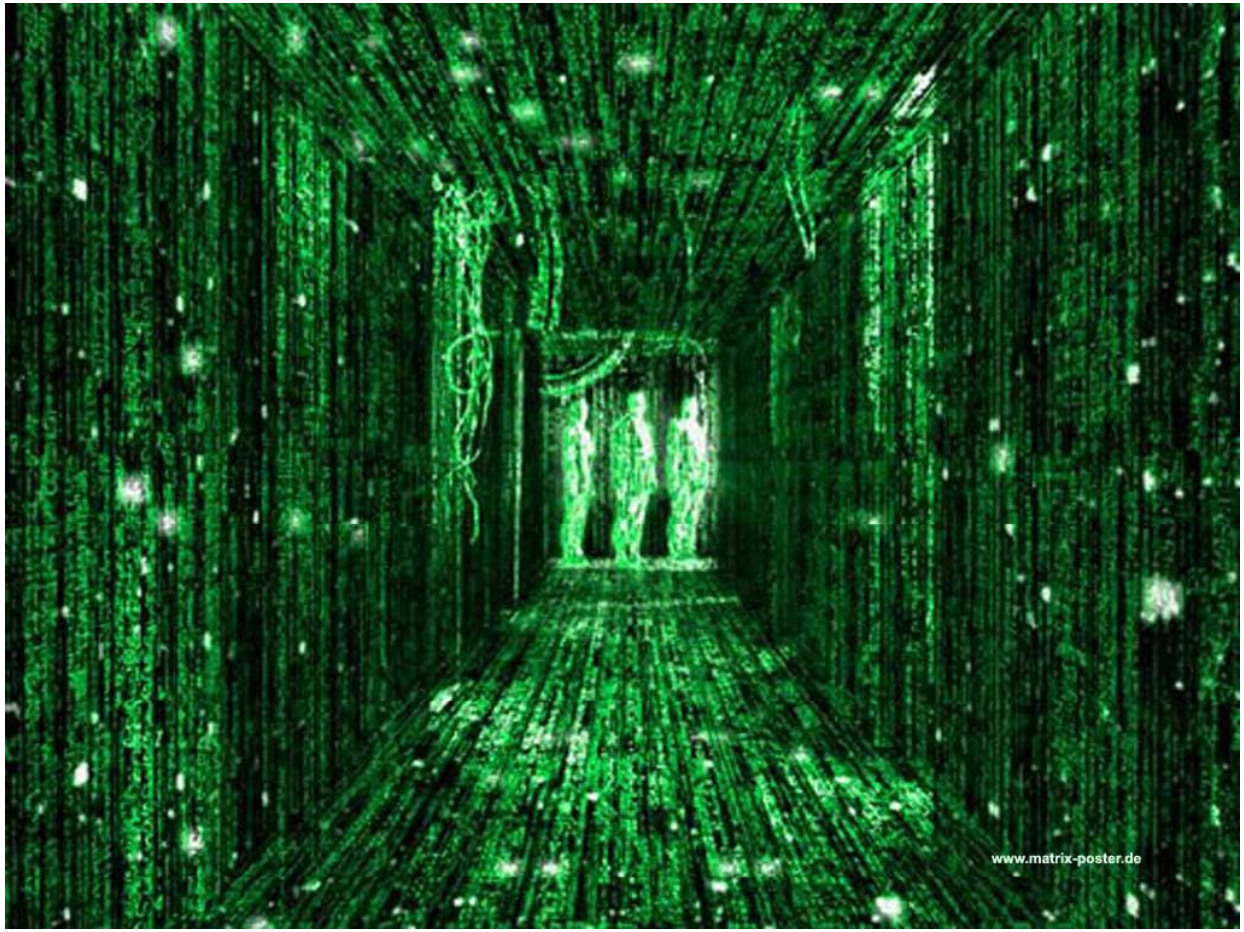
Outline

- **Some definitions**
- **History**
- **Reason and benefits of Virtualization**
- **Virtualization techniques**
- **Hypervisor**
- **Server Virtualization**
- **Approaches for Virtualization**

Source: Wikipedia

Some concepts and history

What is “virtual”?



CERN School of Computing

What is “virtual”?

What is “virtual”?

Definitions @ *Computer Science*

- ***Virtual:*** Created, simulated, or carried on by means of a computer or a computer network
- ***Virtualization:*** Virtualization is the creation of a virtual version of something, such as an operating system, a server, a storage device or network resources.
- ***Virtual Machine:*** A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine.

The beginning

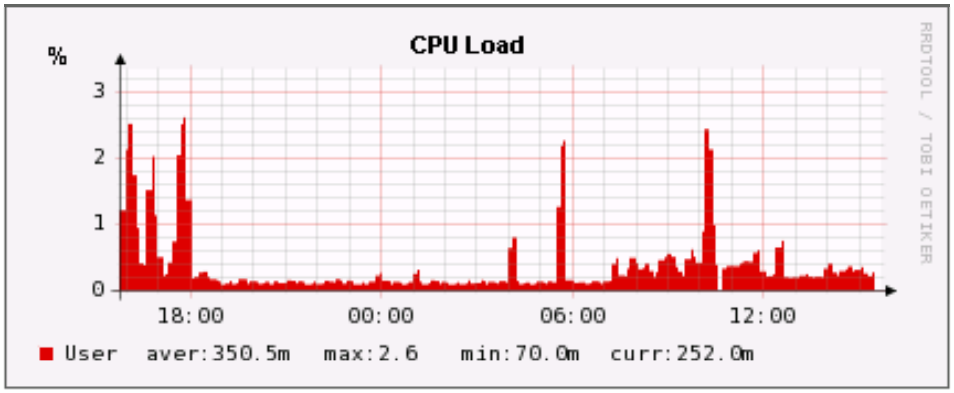
- **1960s**, IBM creates this concept to fully utilize mainframe hardware by logically partitioning them into virtual machines.
- **1980s - 1990s**, desktop computing and x86 servers become available and so the virtualization technology was discarded eventually.
- **1990s**, VMWare invented virtualization for the x86 architecture.
- **Main objective:**
  - Portability of applications
  - Isolation
  - Integration / Deployment (solving dependencies)

Reasons and benefits

Reasons & Benefits of virtualization

- **Low** Infrastructure **Utilization**, more powerful hardware

IDC Average utilization: 10%-15%
Some CERN real cases: <5%



Pre-virtualization machines:
- 8 GB RAM

Virtualization host machines:
- 48 GB RAM

- Decreasing Physical **Infrastructure Costs**

IDC: International Data Corporation, is a market research firm

Reasons & Benefits of virtualization

- **Relocation** from one physical machine to another as needed, helping with **Failover** and **Disaster** Protection

Reality: Power cut December 2010

- Insufficient **Physical Space** in Data Centers

Reality: Our bigger pool: 5 Physical Machines - ~40 Virtual Machines

- High Maintenance end-user desktops
- Server **consolidation**: P2V transformation

Reality: Virtual Machines are identical to Physical Machines, but more flexible
- Installation and configuration
- Usability

Underutilized and wasted resources

More concepts

What can be virtualized?

- **Network virtualization** splits up the available bandwidth into channels, each of which is independent from the others, and each of which can be assigned to a particular server or device in real time.
- **Storage virtualization** is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device. It is commonly used in storage area networks (SANs).
- **Server virtualization** is the idea of taking a physical server and partition it, or divide it up, so that it appears as several “virtual servers”, each of which can run their own copy of an OS. The main purpose of this approach is ISOLATION.

CERN School of Computing

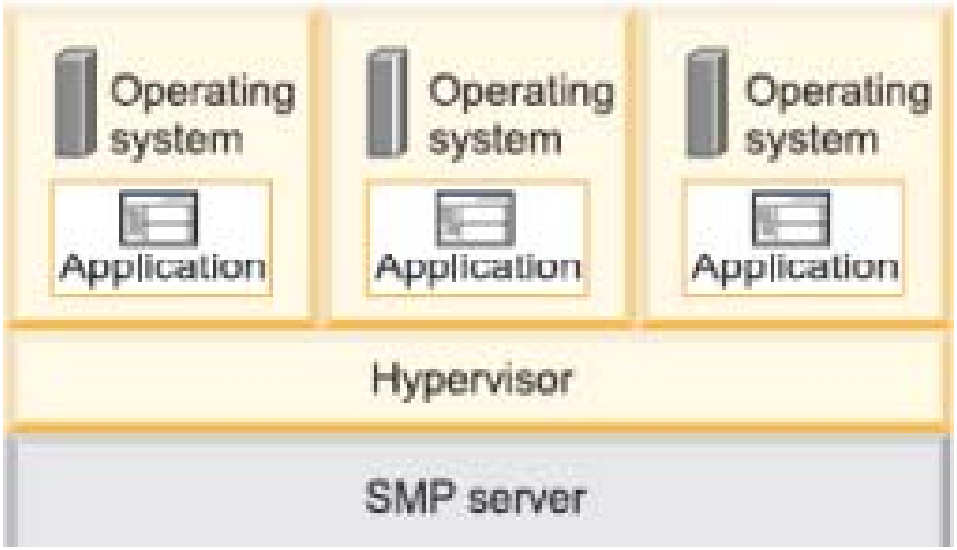
Hypervisor

- Also called **Virtual Machine Monitor (VMM)** or Virtualization Manager
- **Concept:** Software program or part of the code in firmware that manages either multiple OS or multiple instances of the same OS on a single computer system
- What is controlled by the hypervisor:
  - CPU
  - Memory
  - Other resources required by the operating system
- It validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges.

The hypervisor allow to run guests OSes

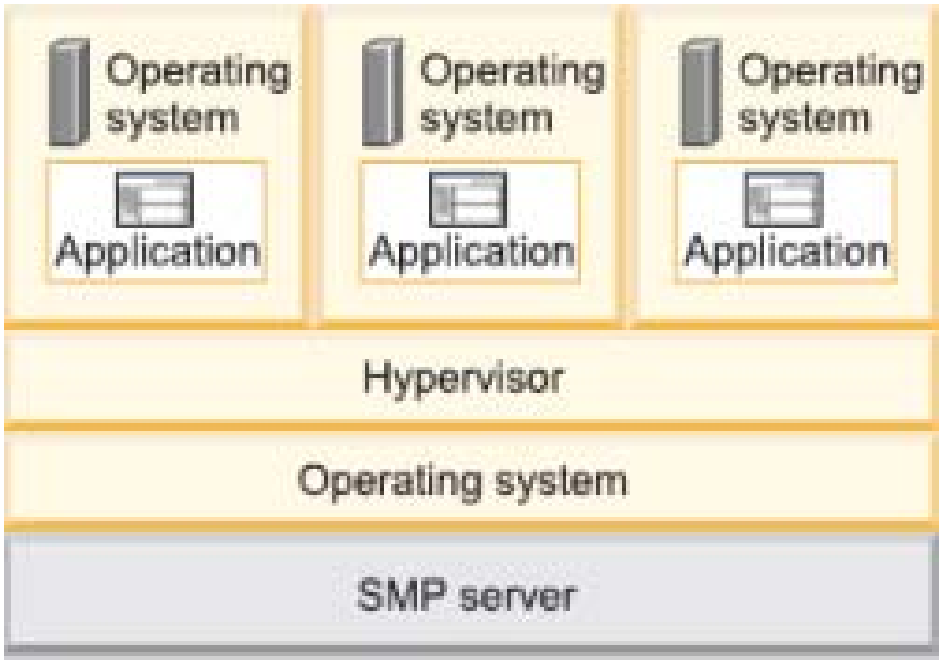
Types of Hypervisor

- **Type 1** hypervisors are those that run directly on the system hardware and offers a higher level of virtualization efficiency and security



Types of Hypervisor

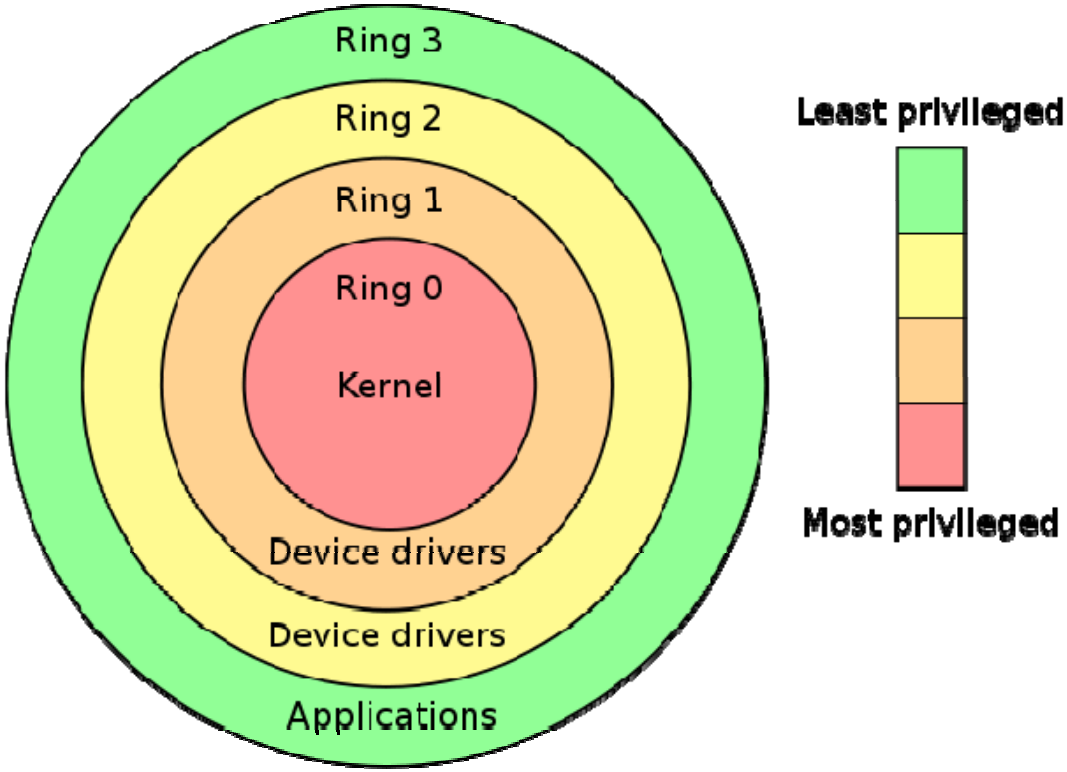
- **Type 2** hypervisors are those that run on a host operating system that provides virtualization services, such as I/O device support and memory management



Needed concept: Protection Rings

Concept: Mechanism to set the layers of privilege used to protect the data and the functionality.

Ring 0: Can execute any CPU instruction and reference any memory address.
Ring 3: The access to hardware and memory reference needs to be arbitrated.



Hypervisor: handles the resource and memory allocation for the virtual machines
Guest's kernels: run in outer rings but they might be written to run in Ring0

SOLUTIONS??

Image source: Wikipedia

Virtualization Techniques

Virtualization Techniques

- **Emulation**
- **Full Virtualization**
- **OS-level virtualization**
- **Paravirtualization**
- **Hardware-assited Virtualization**
- **Application level virtualization**

Emulation

- **Concept:** VM Emulates/simulates complete hardware.
  - Unmodified guest OS for a different PC can be run
  - It is possible to emulate an architecture in a completely different one
- **What runs where?**
  - Ring 0: run the host
  - Ring 3: run the guest OS (as applications)
- **Cons:** Very slow because of not native execution
- **Examples:** Bochs, Virtual PC for Mac, QEMU

Full Virtualization

- **Concept:** The host OS emulates a hardware layer for each guest OS
- **Binary translation technique**
- **What runs where?**
  - Ring 0: run the guest OS privileged operations
  - Hypervisor provides CPU emulation to handle it
- **Pros:** Stability, guest OS doesn't need modifications
- **Cons:** System Resources, time performance
- **Examples:** IBM VM family, VMWare Workstation, Parallels, …

OS-level Virtualization

- **Concept:** Same OS for host and guests and only isolation in the userland.
  - If you run `ps aux` in the host, you will see all guest processes
- **What runs where?**
  - Ring 0: run the OS (same for all)
- **Pros:** Low overhead, highest performance
- **Cons:** Isolation, stability
- **Examples:** FreeBSD Jails, Solaris Containers, Virtuzzo/OpenVZ

Paravirtualization (I)

- **Concept:** The hypervisor, interfaces the hardware to all OS's (host and guest)
  - Guests are modified to run on the hypervisor
    - Replacement of privileged operations (ring0) with calls to the hypervisor

- **What runs where?**
  - Ring 0: run an hypervisor
  - Ring 1: run the guest OS
  - Ring 3: run the applications

Paravirtualization (II)

- **Pros:**
  - **Stability** is very good
  - **Performance** is very good (nothing can beat OS-level virtualization in this matter)
  - **Overhead** is very **low**
- **Cons:**
  - **Not easy to implement** (it's getting better these days)
  - Both **host** and **guest kernels** has to be **patched**
  - **Maintainability**
- **Example:** XEN, VMWare, ESX server

Hardware-assisted Virtualization (I)

Sometimes it's called accelerated virtualization or HVM (Hardware Virtual Machines)

- **Concept:** Hardware provides support to run instructions independently for each OS
  - Facilitates the IN-OUT privileged mode
  - Support for address translation
- **What runs where?**
  - Ring 0: run the guest OS privileged operations
  - Latest generation CPUs provide built-in features to run unmodified guest OS without overhead
- **Implementation:** IBM 370 (1972), Intel VT, AMD-V, UltraSparc and others
- **Examples:** Linux KVM, VMware fusion, Microsoft Virtual PC, XEN, Virtual Box

Hardware-assisted Virtualization (II)

- **Pros:**
  - **No** need to **patch** the guest OS
  - Most **optimal performance**
  - **Great Stability**
- **Cons:**
  - Need to be **supported** by the **hardware**

Application level virtualization

- **Concept:** Application is given its own copy of components that are not shared (e.g. own registry files, global objects) to prevent conflicts
- **Features**
  - Can run only a single process
  - VMs are implemented using an interpreter
- **What runs where?**
  - Ring 3: The VM run in this level
- **Examples:** JVM

Comparison table

	Performance	Stability	Isolation	Modifications? / HW support?
Emulation	↓ -	↓	↑	**No/No**
Full Virtualization	↓	↑	↑	**No/No**
OS-level virtualization	↑	↓	↓	**No/No**
Paravirtualization	↑+	↑	↑	**Yes/No**
Hardware-assited Virtualization	↑+	↑	↑	**No/Yes**
Application level virtualization	↓	↑	↑	**No/No**

Server Virtualization

Server Virtualization

- **Concept:**
  - Is the idea of taking a physical server and partition it, or divide it up, so that it appears as several “virtual servers”, each of which can run their own copy of an OS.

Why is it so popular?

- **Multiple operating system** instances at once on a **single physical server**.
- **Reduce** the **number** of **physical servers** and in consequence physical space in data centers
- Offers **higher availability**
- **Cut down** on **energy** consumption
- Eases the **maintenance** and **patching**

Server Virtualization Landscape

- As of October **2009**, only **16 %** of server workloads were running **on VMs**
- Gartner predicted that the numbers would rise to approximately **50 %** of workloads (x86 architecture) by the **end of 2012** – or about 58 million deployed machines

Highlighted benefits

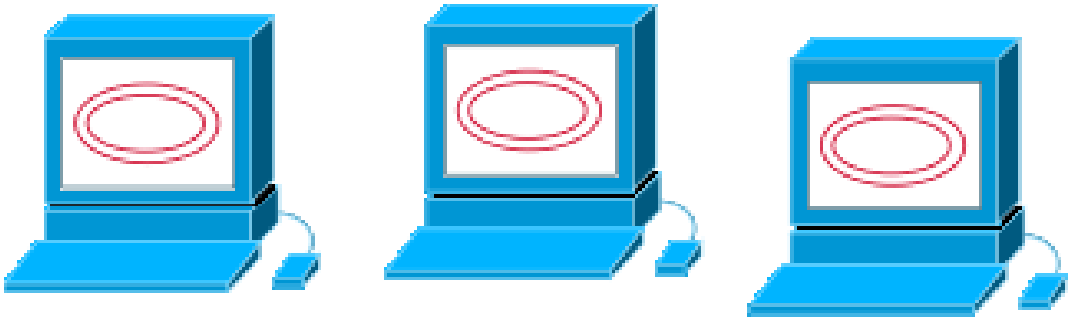
Source: Gartner Inc.

How we do it at CERN

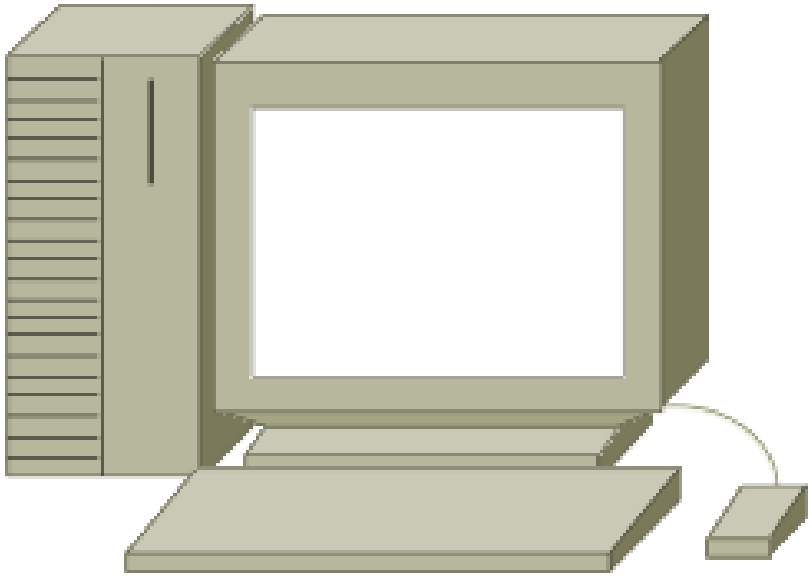
- The idea of **P2V** (Physical to Virtual)
- Management via **Quattor** (Extremely Large Fabric Management System)
- **Not golden image** distribution, every VM is like a physical machine

Practical Approaches

Simple Virtualization ☺

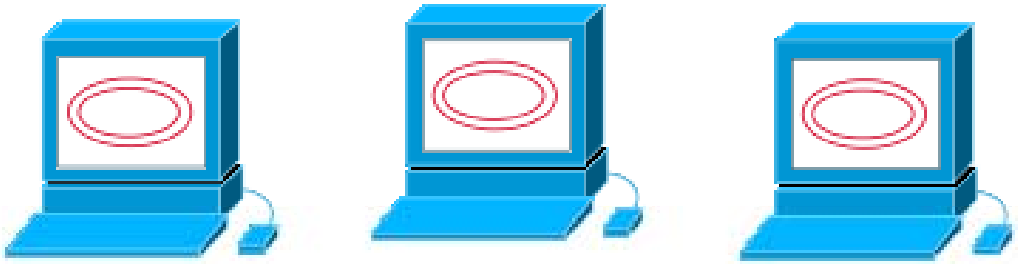


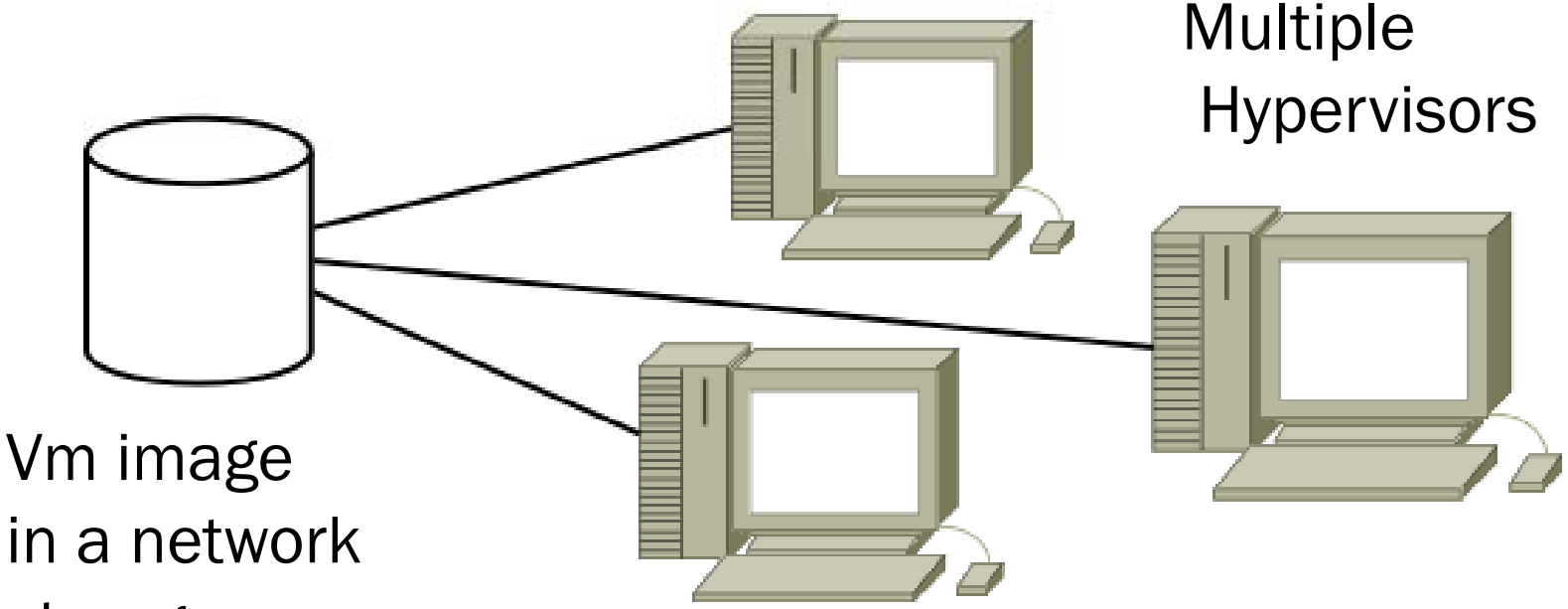
Virtual Machines



Physical Hypervisor

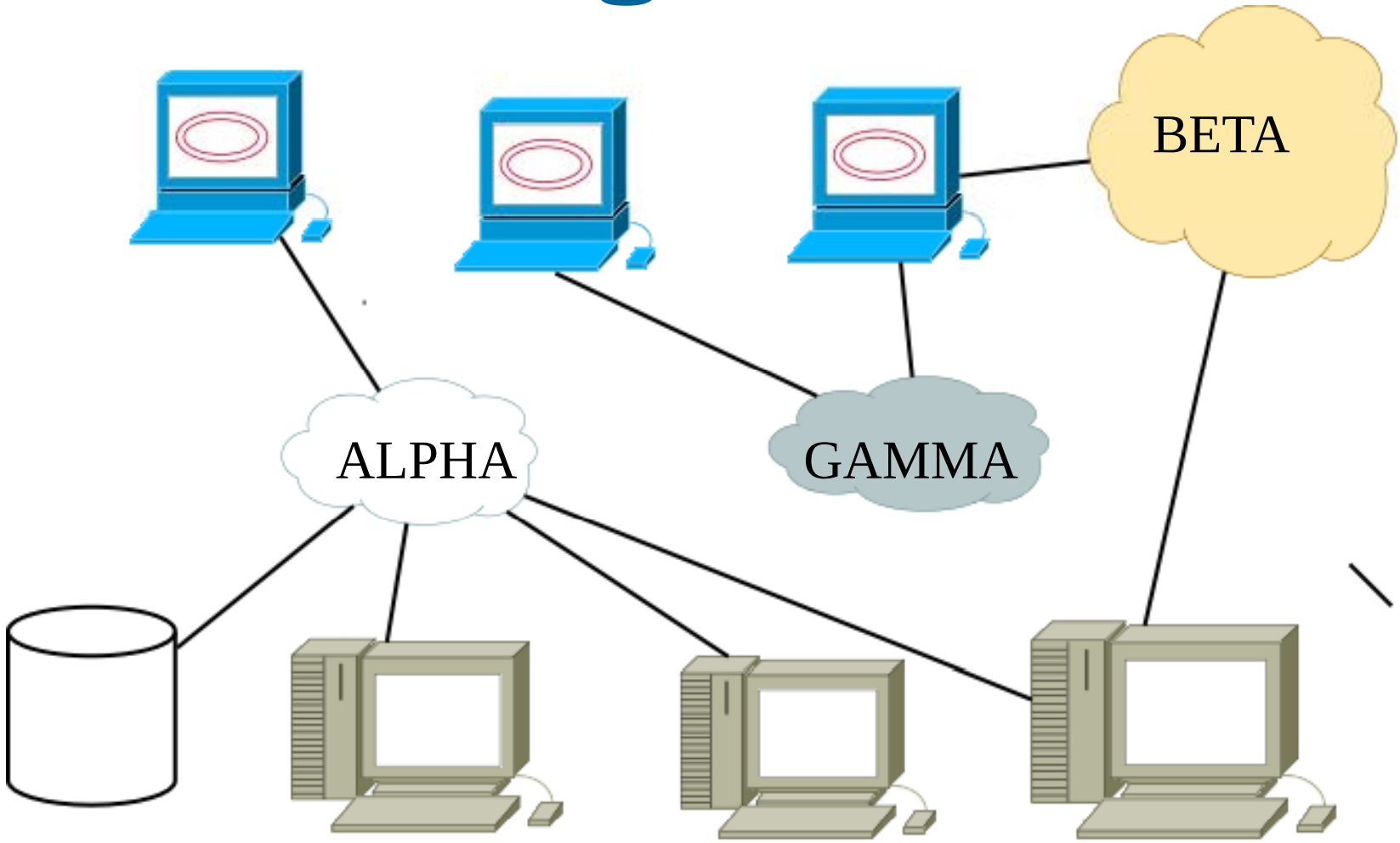
More virtual approach ☺



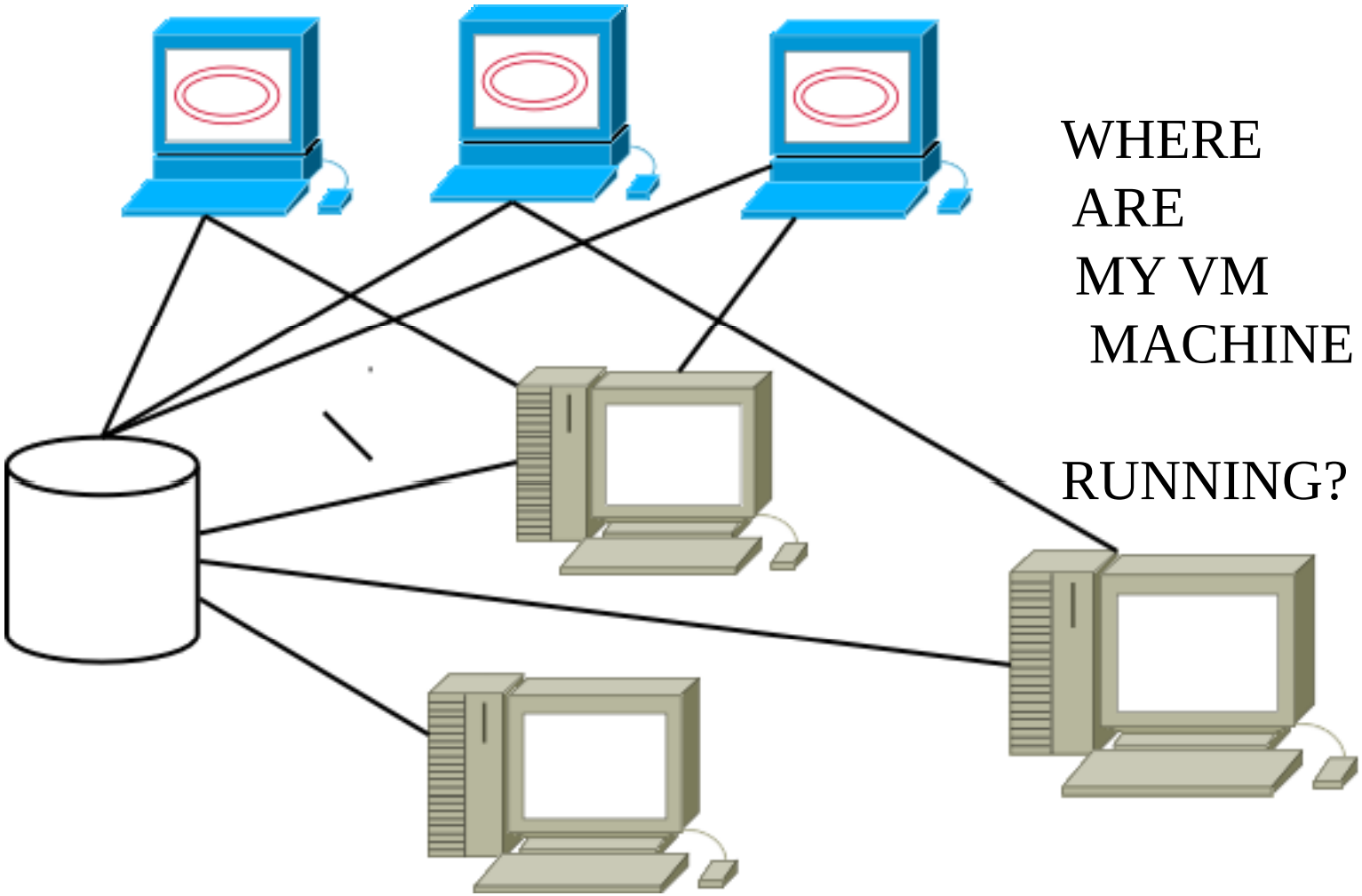


CERN School of Computing

Virtual networking



Management scenario ☹

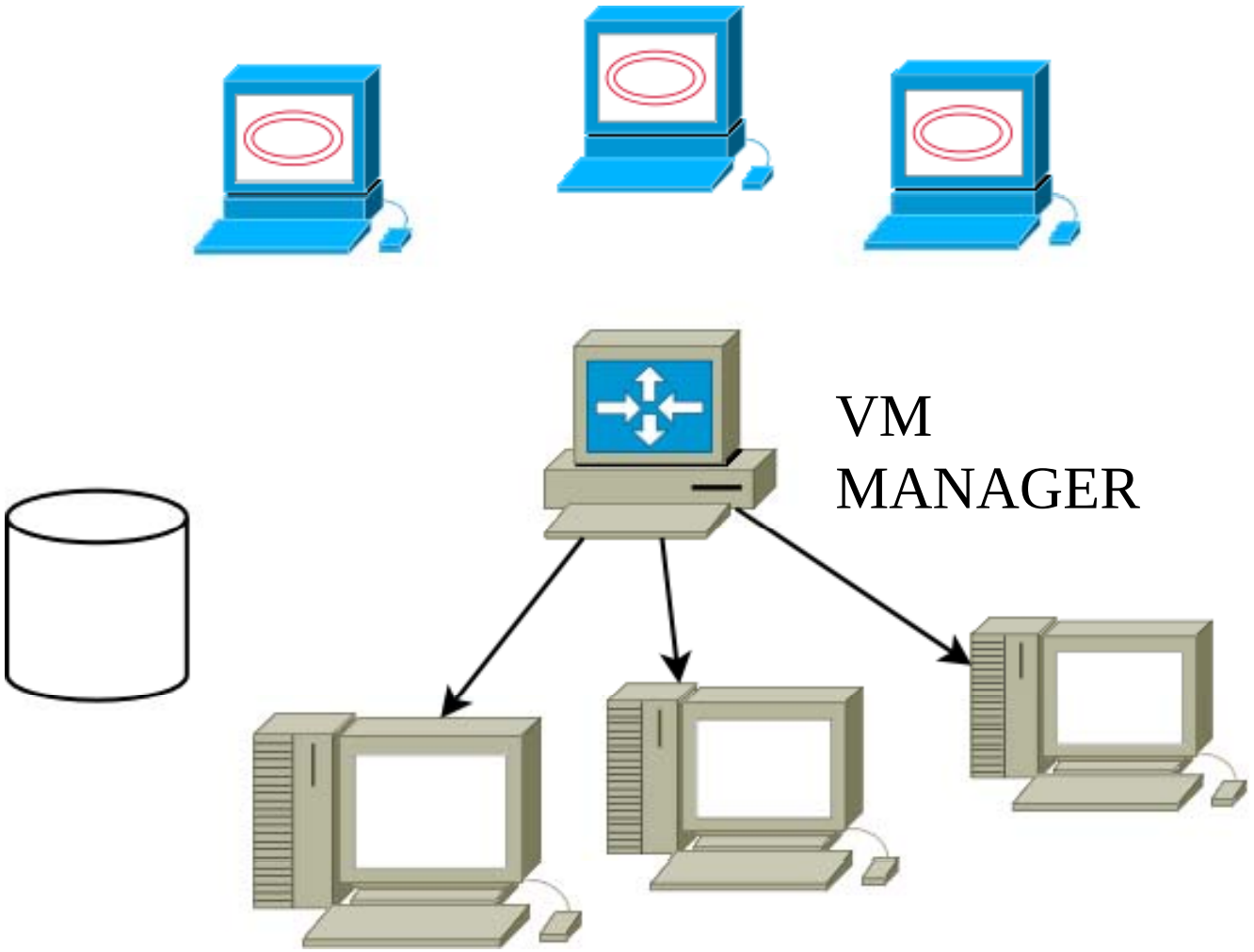


WHERE
ARE
MY VM
MACHINE

RUNNING?

CERN School of Computing

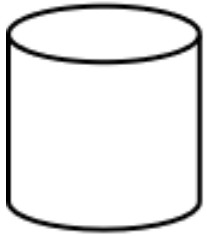
VM Manager ☺

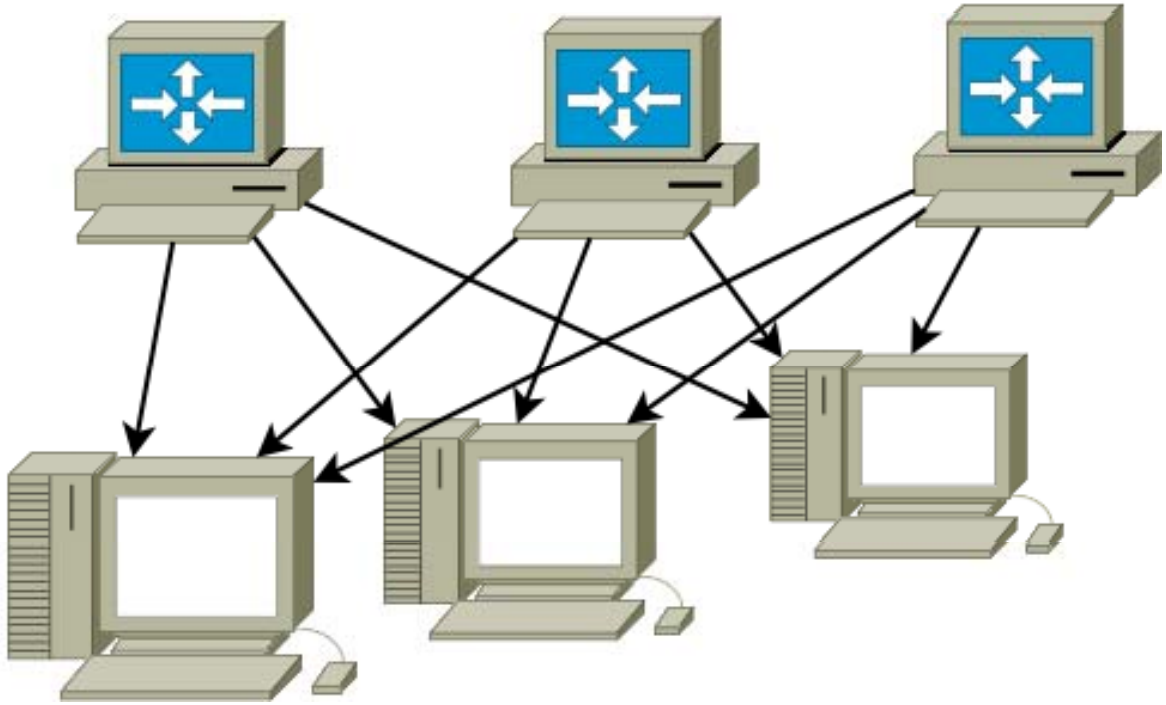


CERN School of Computing

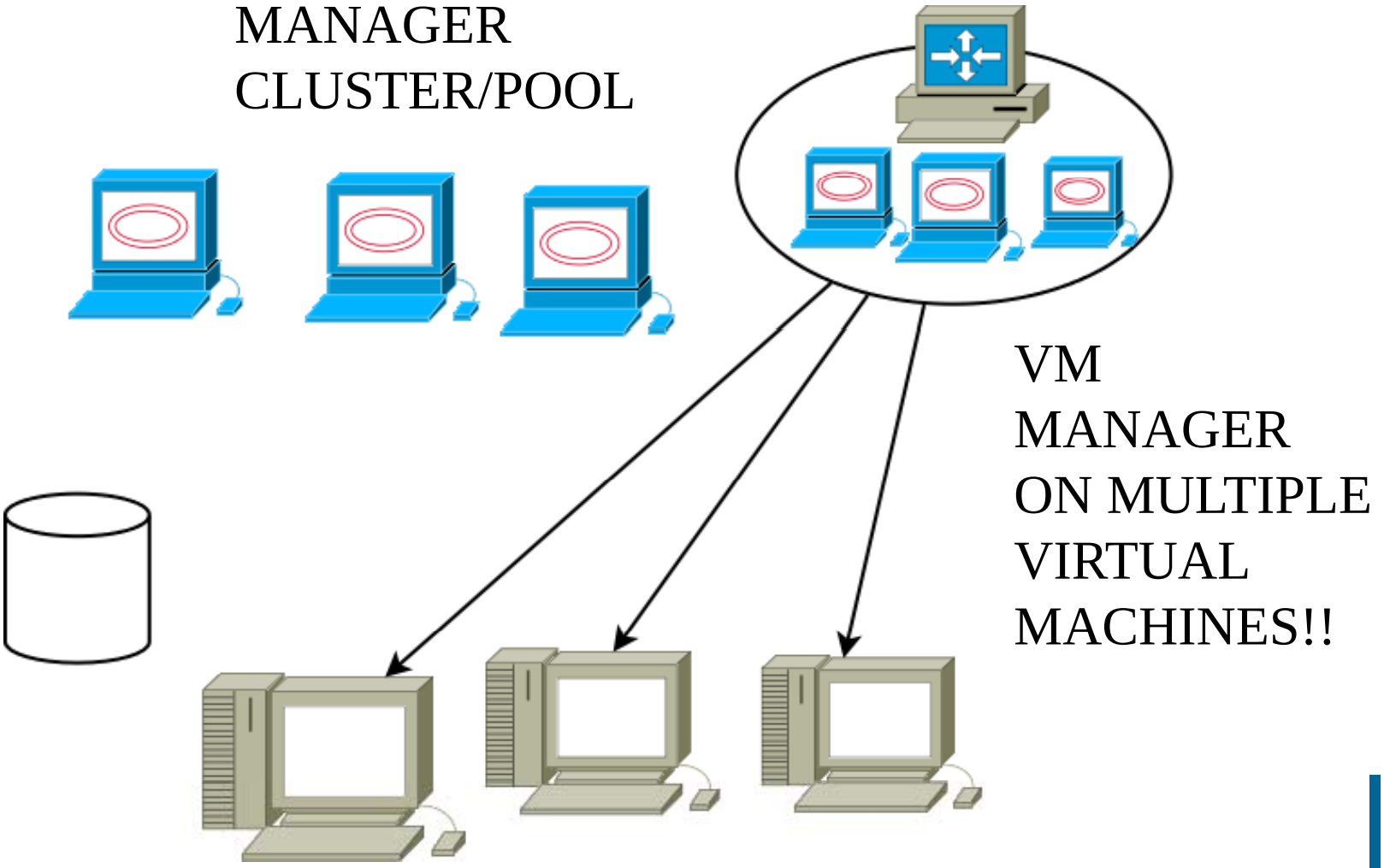
More virtual approach ☺☺

MULTIPLE
VM
MANAGERS



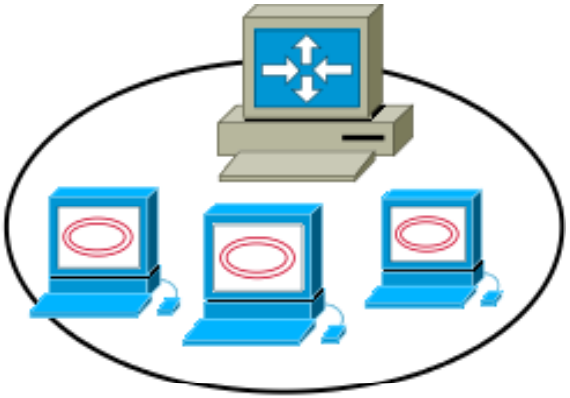


Full virtual approach

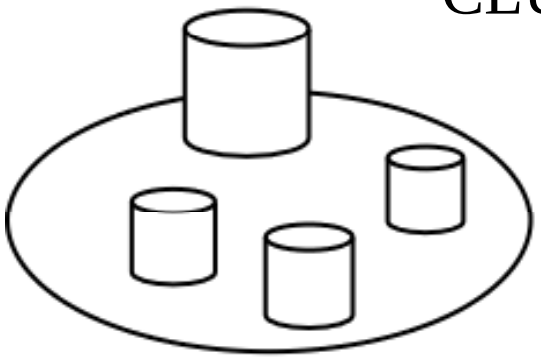


Resource clusters/pools

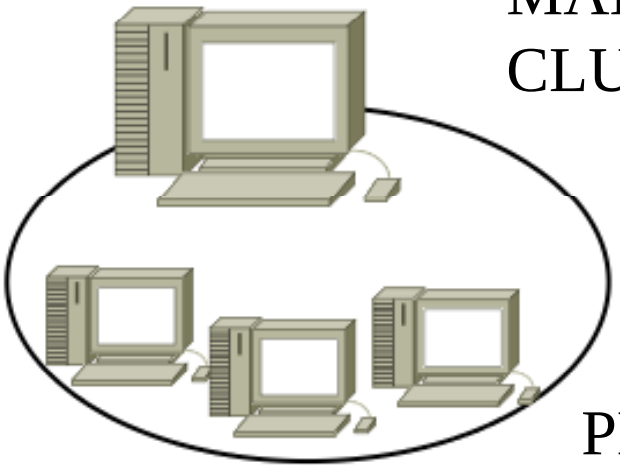
VM POOL CLUSTER



VM MANAGER CLUSTER

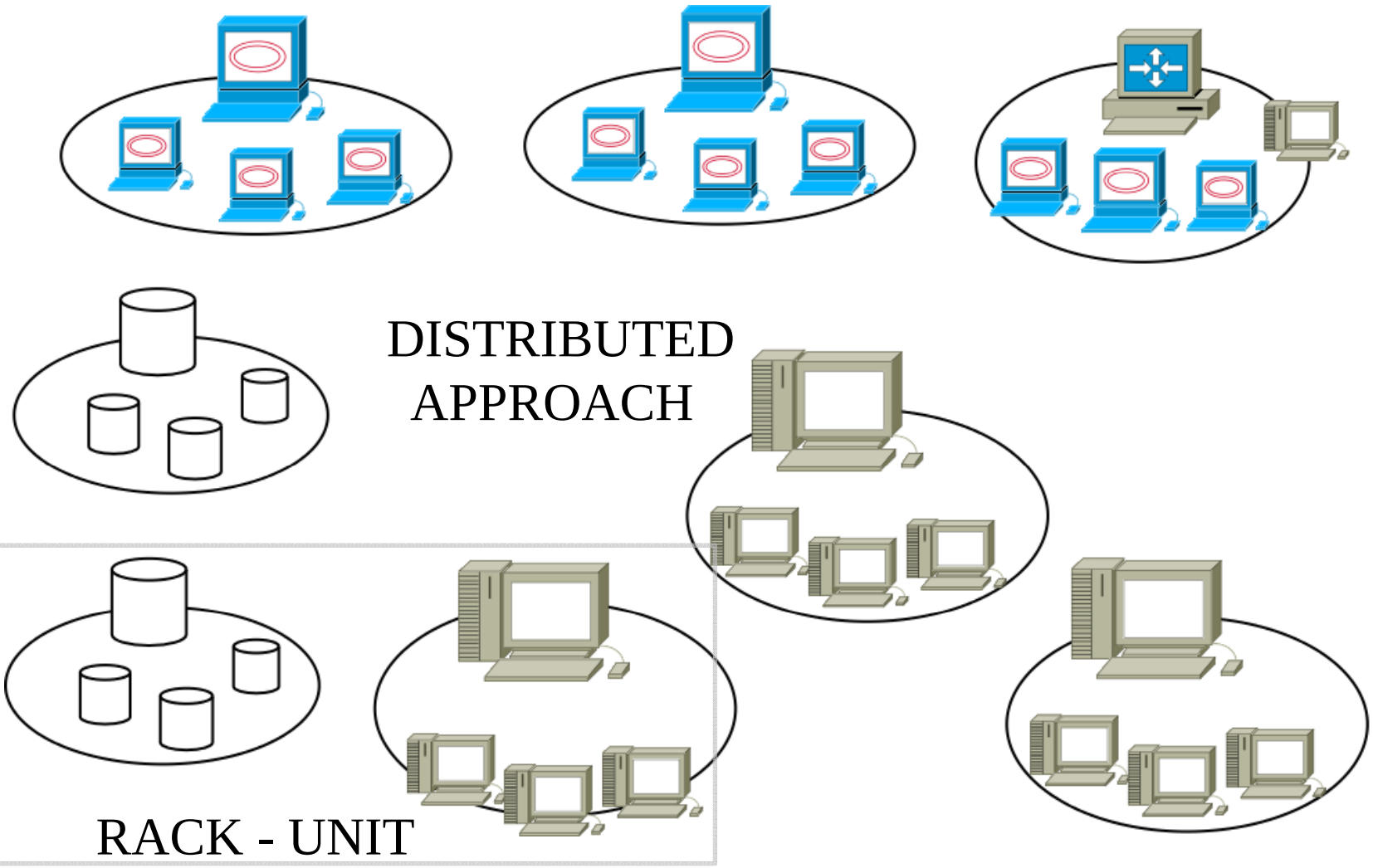


STORAGE CLUSTER



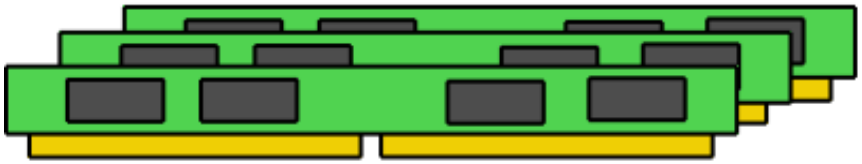
PHYSICAL CLUSTER

More virtual again!

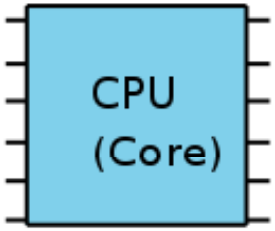


Virtualization ingredients

~~RUNNING~~ EXISTING VIRTUAL MACHINES



MEMORY

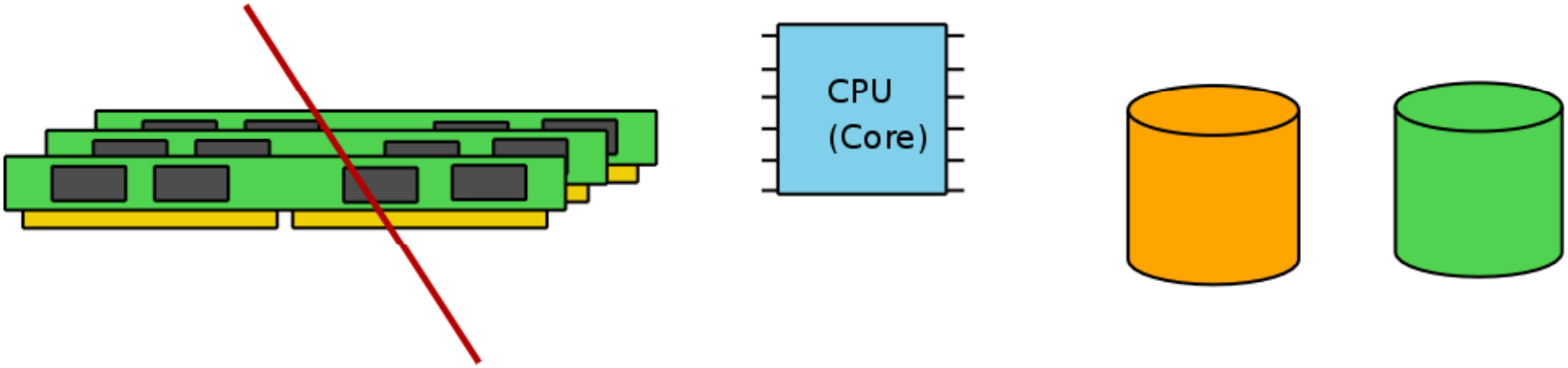


CPU CORE
Better if V optimized

STORAGE

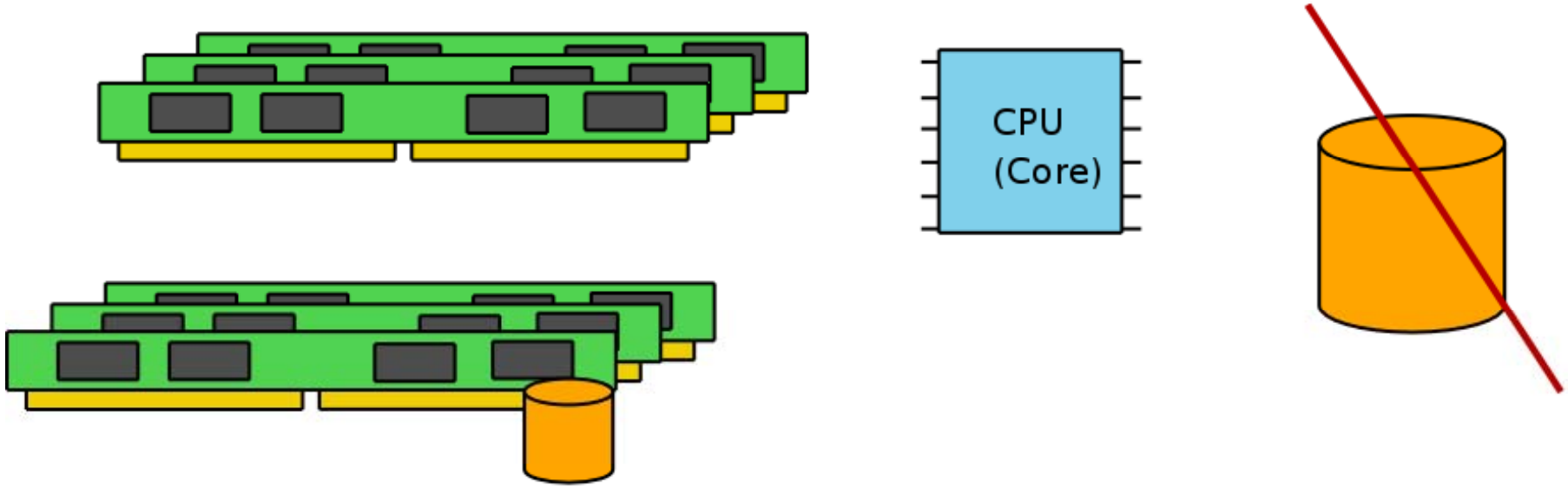
CERN School of Computing

Running VM with no memory!



We can swap all the VM memory in a physical disk, or we use the storage as a slow virtual memory

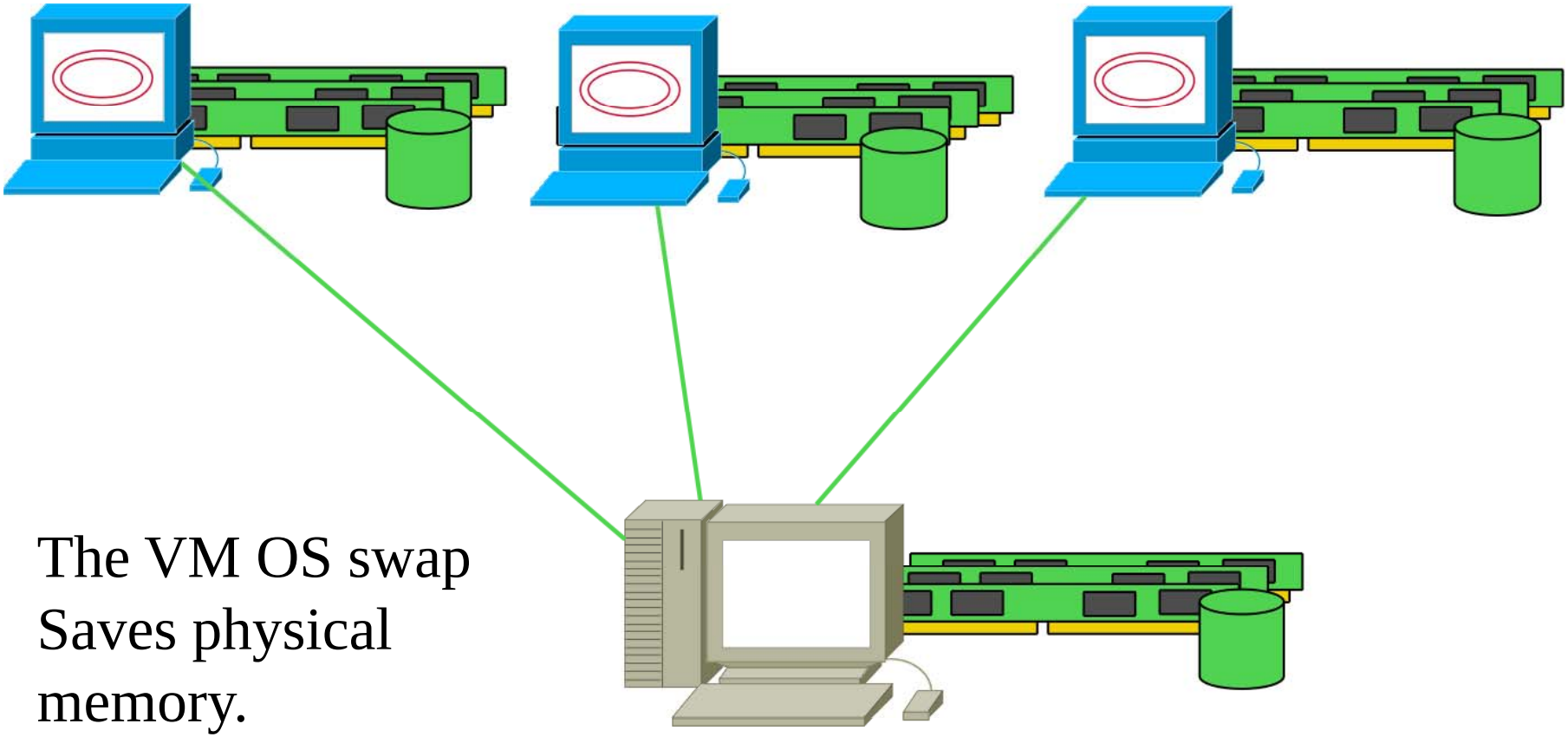
Running VM with storage!



We could use the memory as a physical storage (tmpfs) for the Vm image. Extremely fast virtual machine!

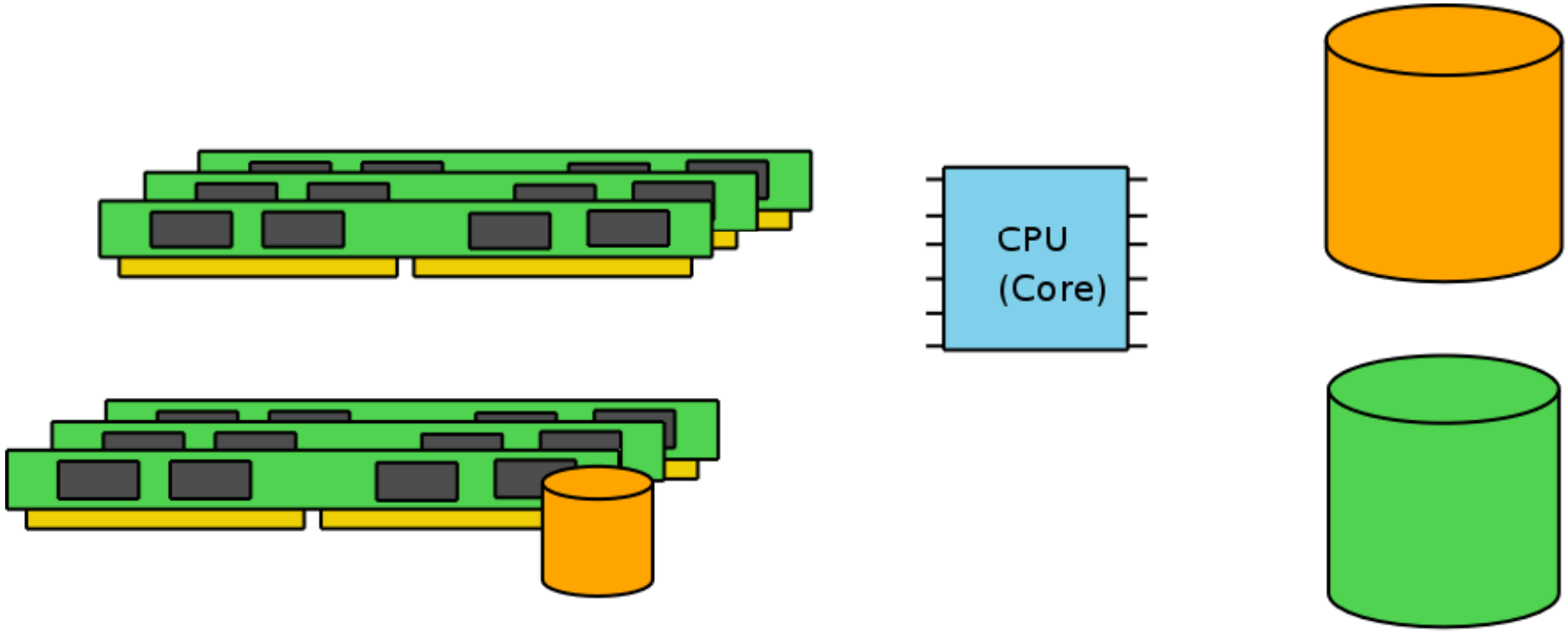
CERN School of Computing

Memory Ballooning



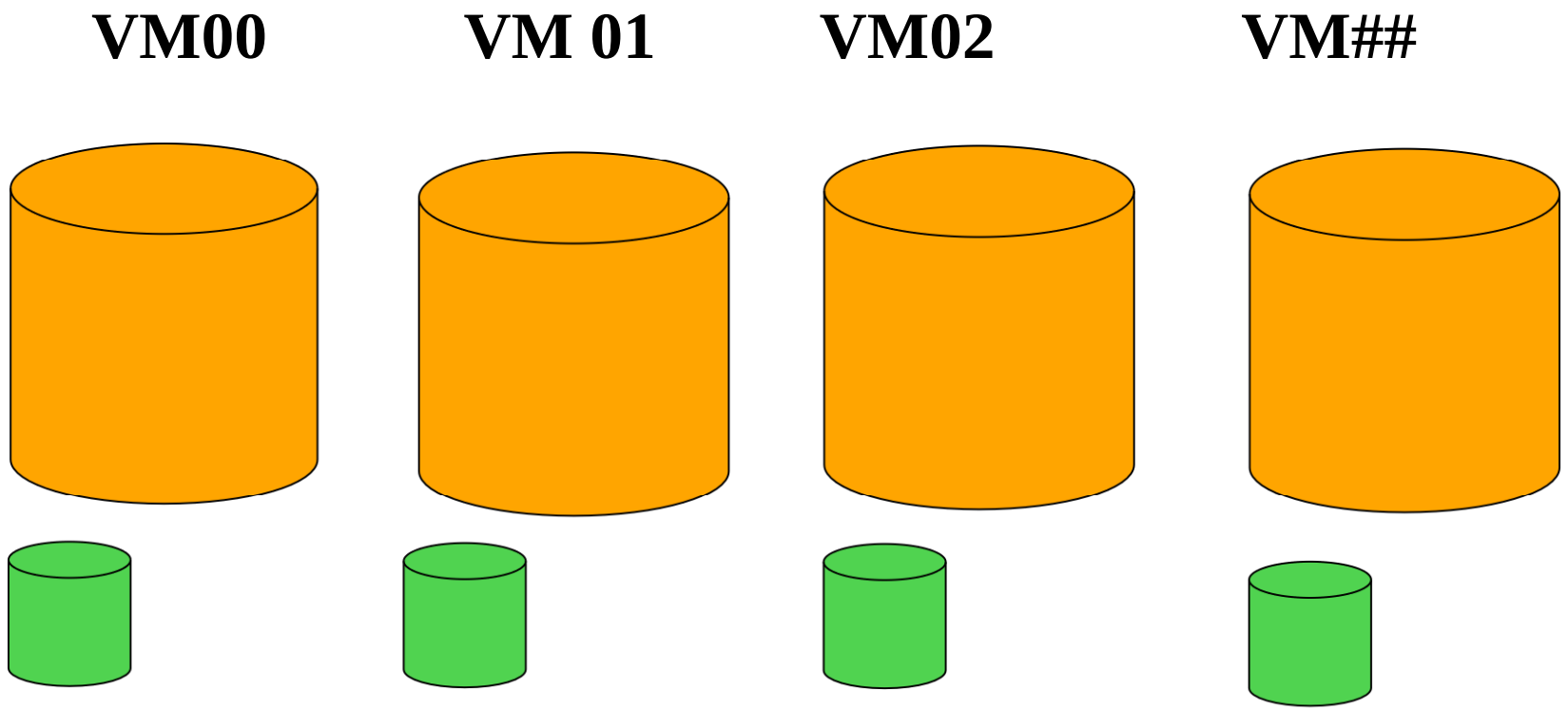
The VM OS swap
Saves physical
memory.
Controlling it we can have more Vmem than PMem

The full virtual approach

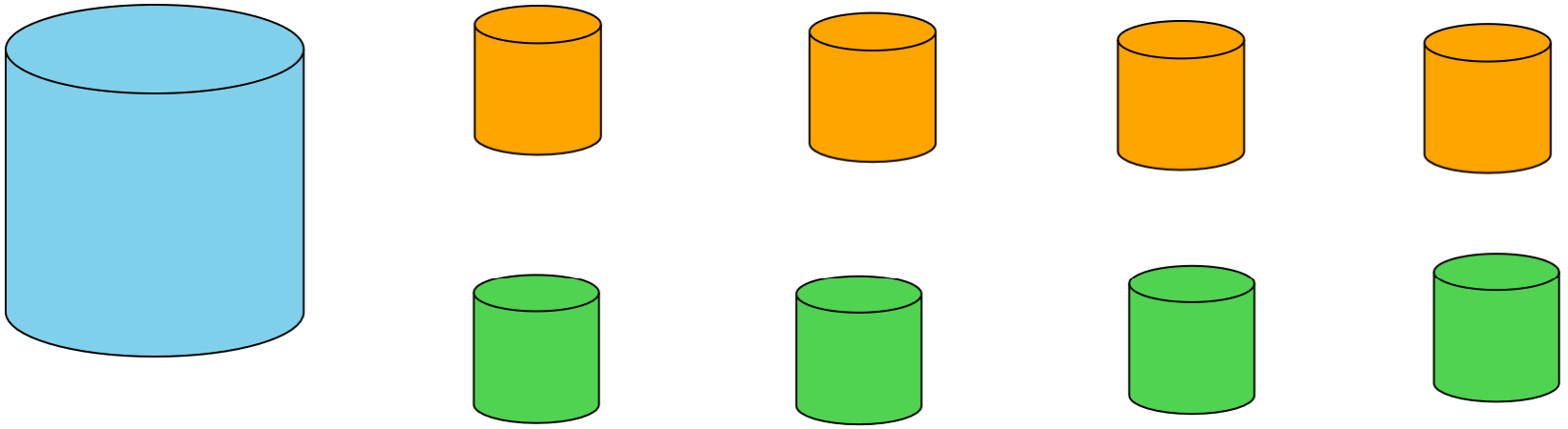


We use all the possible approaches in order to obtain what we want. The memory is really Virtual

Disk image scenario

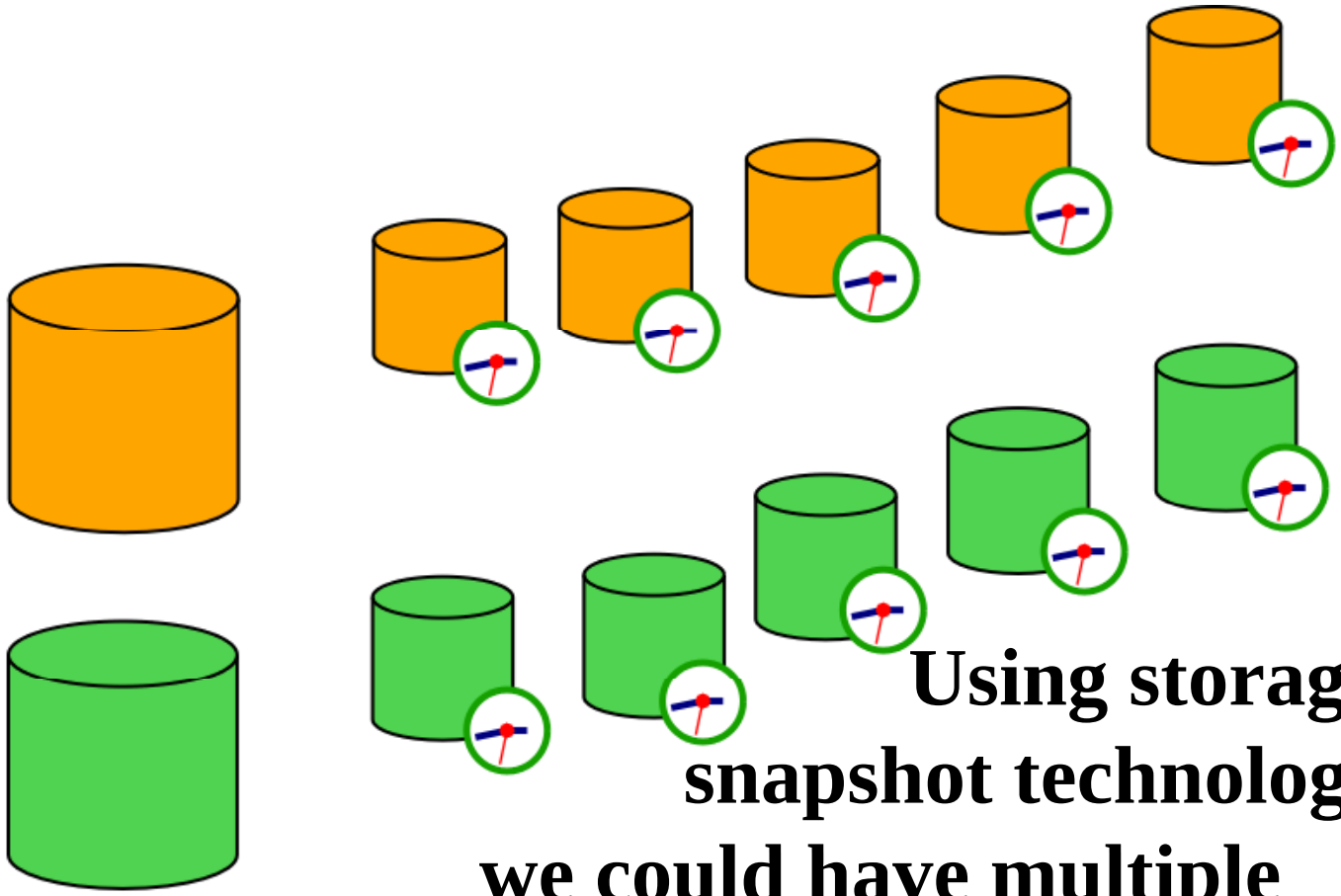


Disk image optimization



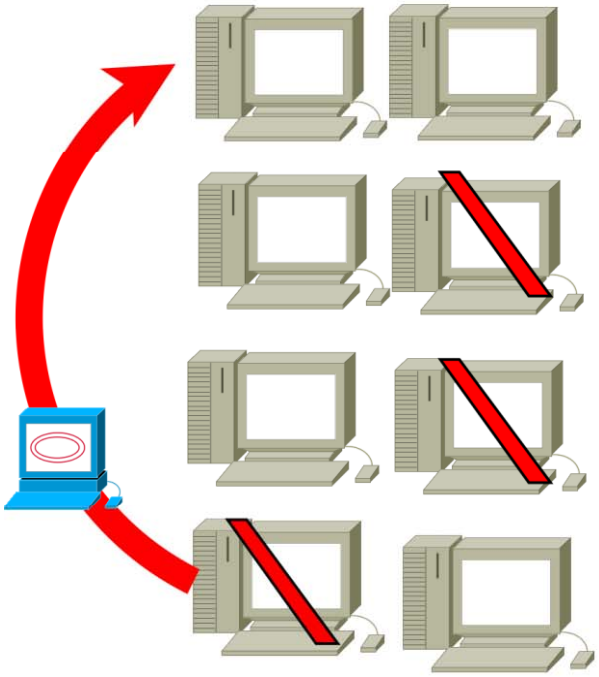
BIG READ ONLY SHARED DISK
Smaller RW disk images

VM versions, time machine!

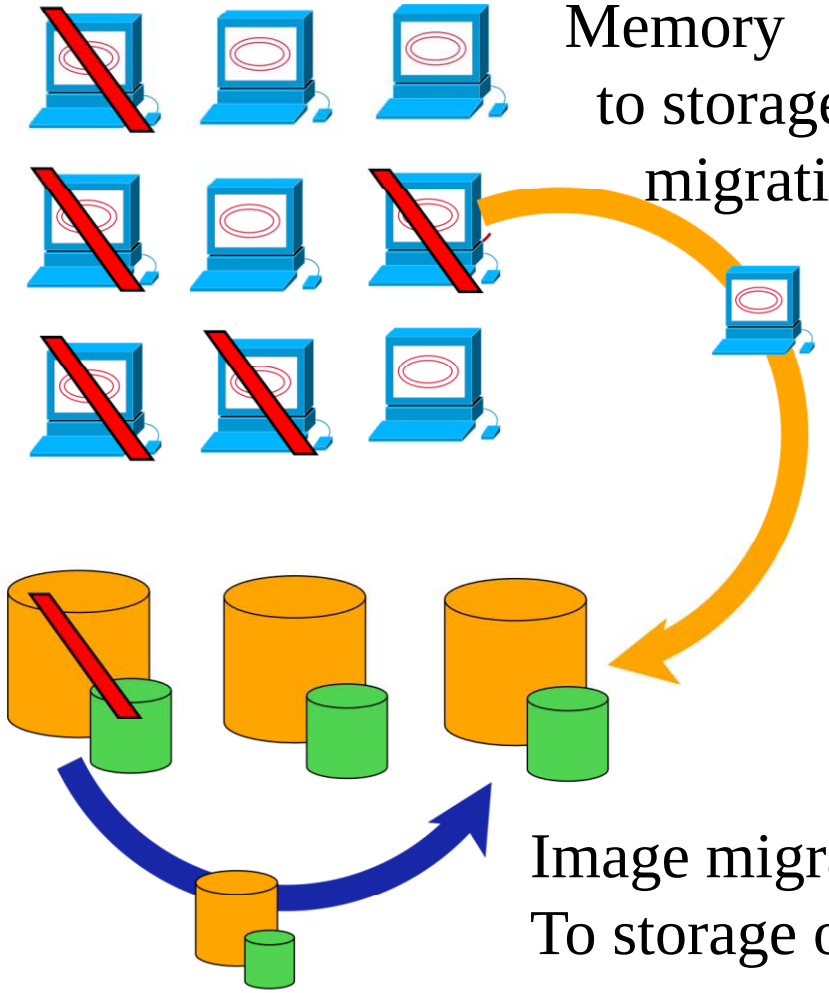


Using storage snapshot technology we could have multiple version of the same VMachine

VM Migrations



VM Migration from One physical host to another one. LIVE!!

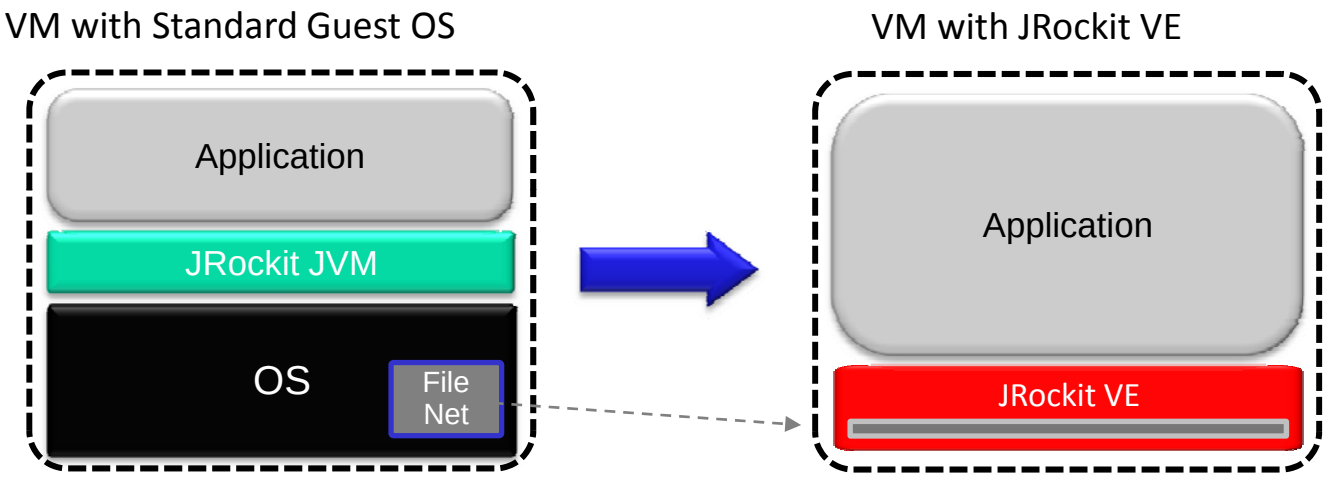


Memory to storage migration

Image migrations, To storage or memory

What about running without OS?

JRockit VE: Removing the OS and Creating a More Efficient Software Stack



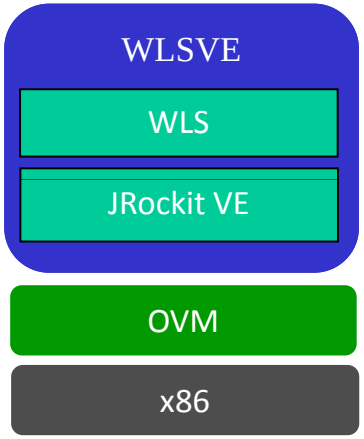
- ~1GB -> ~2 MB
- Improved performance
- Simplified configuration
- Increased security

- **Customized to run single Java process**
- **No shell access allowed**
- **Headless**

Slide from "Oracle JRockit – What's new and what's coming" @ OOW2009 © 2009 Oracle Corporation

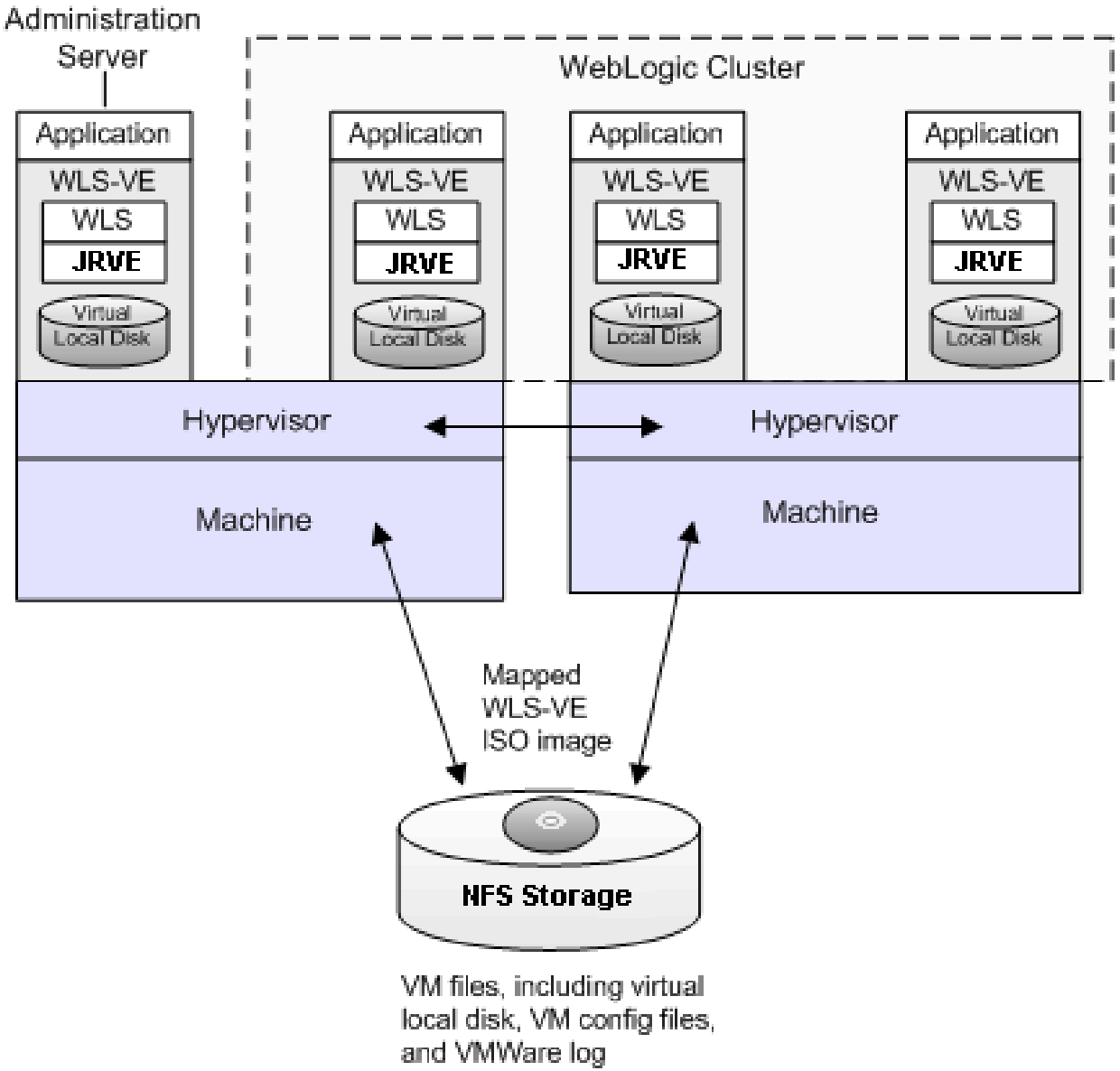
WebLogic Server Virtual Edition: Product Taxonomy

- WebLogic Server Virtual Edition
  - Virtual machine **containing WLS and JRockit VE**
  - Designed to run on Oracle VM, **without an operating system**
  - Users can create their own virtual machine images containing WLSVE and their domains and applications
- JRockit VE
  - **JRockit VE** is the JRockit JVM extended so it **can run directly on virtual hardware**, and optimized for running Java on OVM and x86 hardware
- JRVE Image Tool
  - Create and edit the virtual machine images

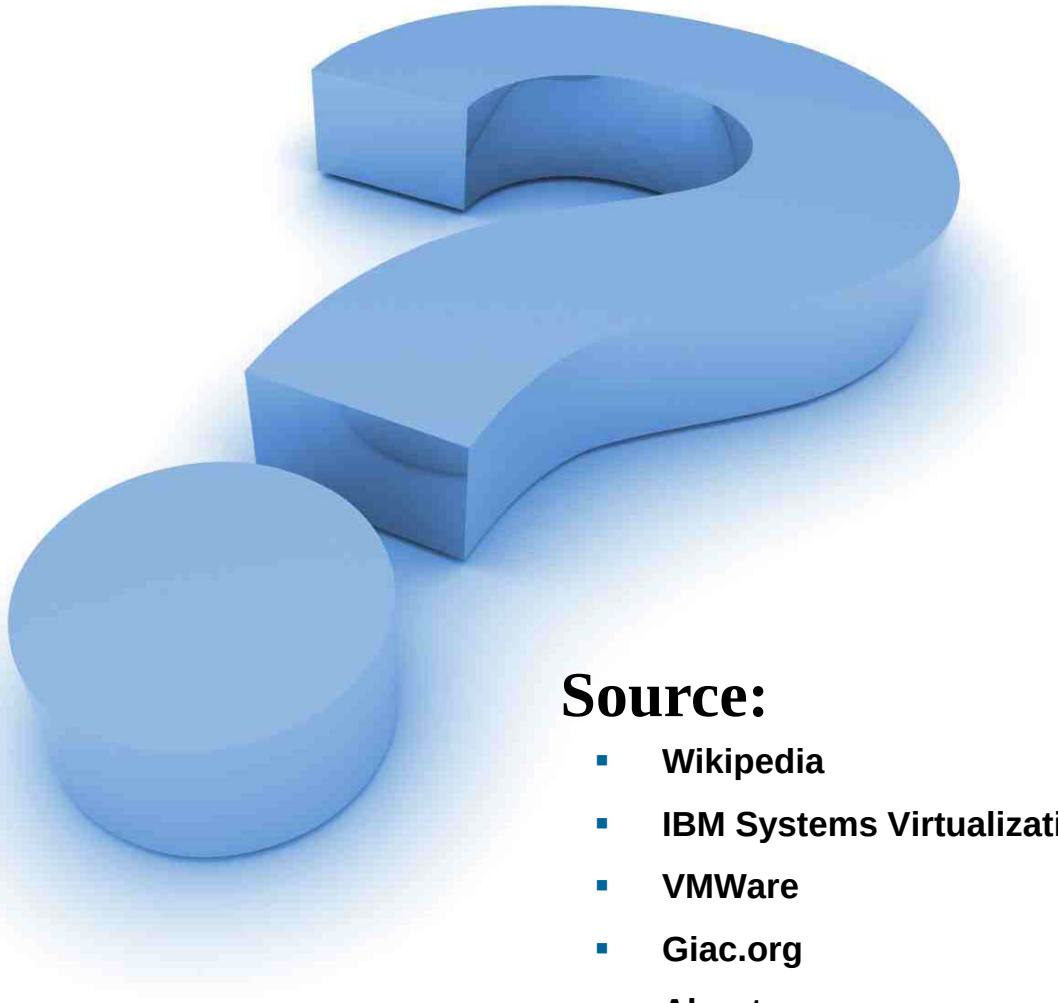


Slide from "Oracle JRockit – What's new and what's coming" @ OOW2009 © 2009 Oracle Corporation

Example of WLS-VE topology



Questions?



Source:

- **Wikipedia**
- **IBM Systems Virtualization Paper**
- **VMWare**
- **Giac.org**
- **About.com**
- **Oracle.com**