IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CLOUD COMPUTING

SOMMAIRE

1. INTRODUCTION
2. CLOUD PRINCIPLES
3. DATA CENTER -INFRASTRUCTURE
4. INSIDE THE CLOUD MODEL
5. INFRASTRUCTURE AS A SERVICE
6. SOFTWARE AS A SERVICE
7. CONCLUSION

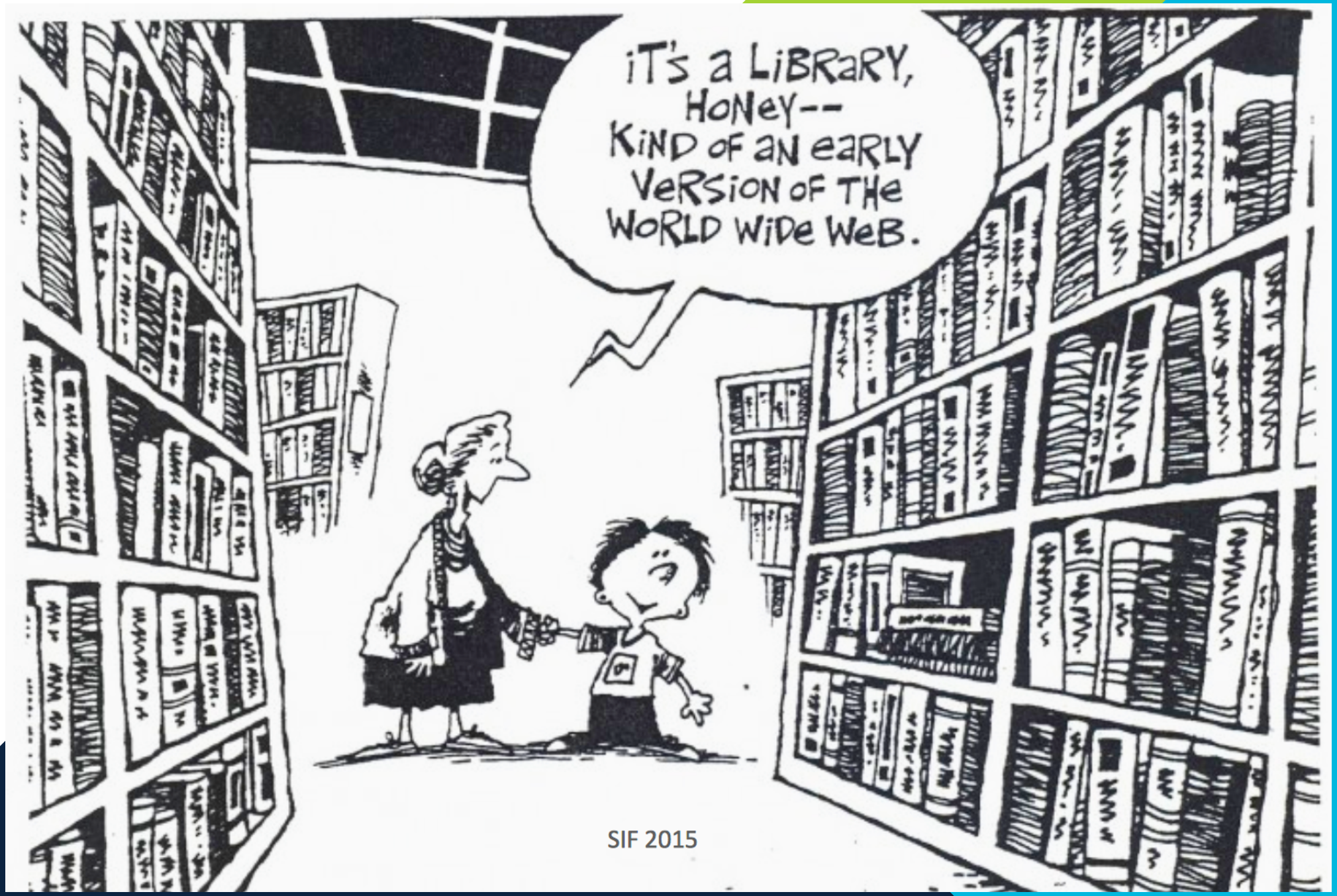
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

INTRODUCTION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

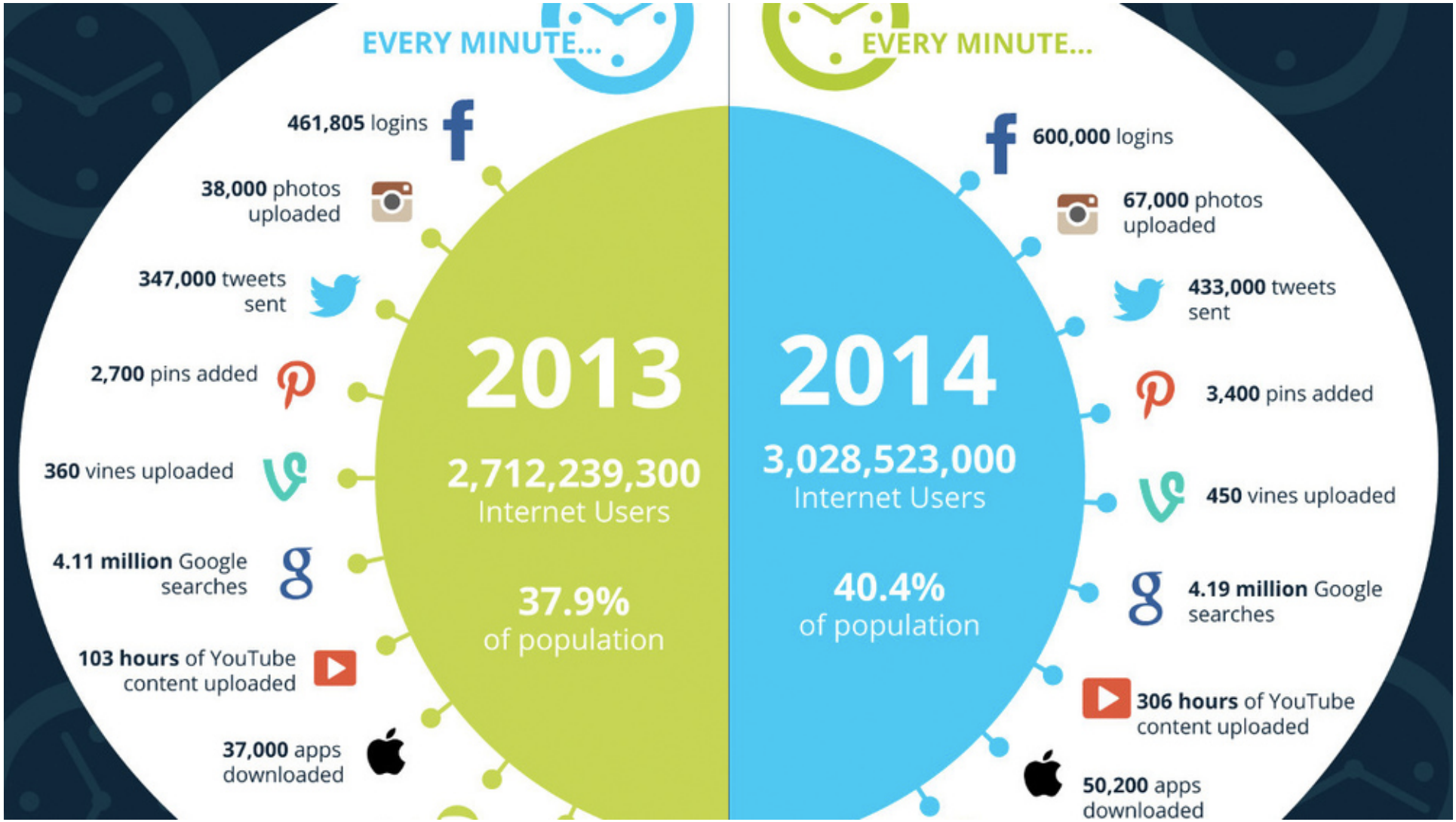
Jean-Marc Menaud

- **Full 1st Class Professor at IMT Atlantique**
  - Energy consumption optimization in large-scale distributed system
  - **Data Center, Virtualization, Dynamic Consolidation, Workload driven, Power driven…**
- **Co-Founder of EasyVirt**
  - French SME on Management solution for virtualized Data Center.


iT's a LiBRaRY, HoNey-- KiND of aN eaRLY VeRSioN of THe WoRLD WiDe WeB.

SIF 2015

EVERY MINUTE



CHAPTER 1
CLOUD PRINCIPLES

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

WHAT IS CLOUD COMPUTING?

Cloud Computing is a general term used to describe a new class of network based computing that takes place over the Internet,

- basically a step on from Utility Computing
- a collection/group of integrated and networked hardware, software and Internet infrastructure (called a platform).
- Using the Internet for communication and transport provides hardware, software and networking services to clients

These platforms hide the complexity and details of the underlying infrastructure from users and applications by providing very simple graphical interface or API (Applications Programming Interface).

WHAT IS CLOUD COMPUTING?

- In addition, the platform provides on demand services, that are always on, anywhere, anytime and any place.
- Pay for use and as needed, elastic
  - scale up and down in capacity and functionalities
- The hardware and software services are available to
  - general public, enterprises, corporations and businesses markets

@Mark Baker

WHAT IS CLOUD COMPUTING?

Cloud computing is an umbrella term used to refer to Internet based development and services

A number of characteristics define cloud data, applications services and infrastructure:

Remotely hosted: Services or data are hosted on remote infrastructure.

Ubiquitous: Services or data are available from anywhere.

Commodified: The result is a utility computing model similar to traditional that of traditional utilities, like gas and electricity - you pay for what you would want!

CLOUD A GOOD OR BAD IDEA ?

In 1997, Steve Jobs introduced his personal cloud based data management system and the efficiency of accessing data from any computer in the world. He outlines a vision for the future where cloud based computing will create safe systems free of hard drive failures and loss of data.

During a WWDC Q&A session with developers.

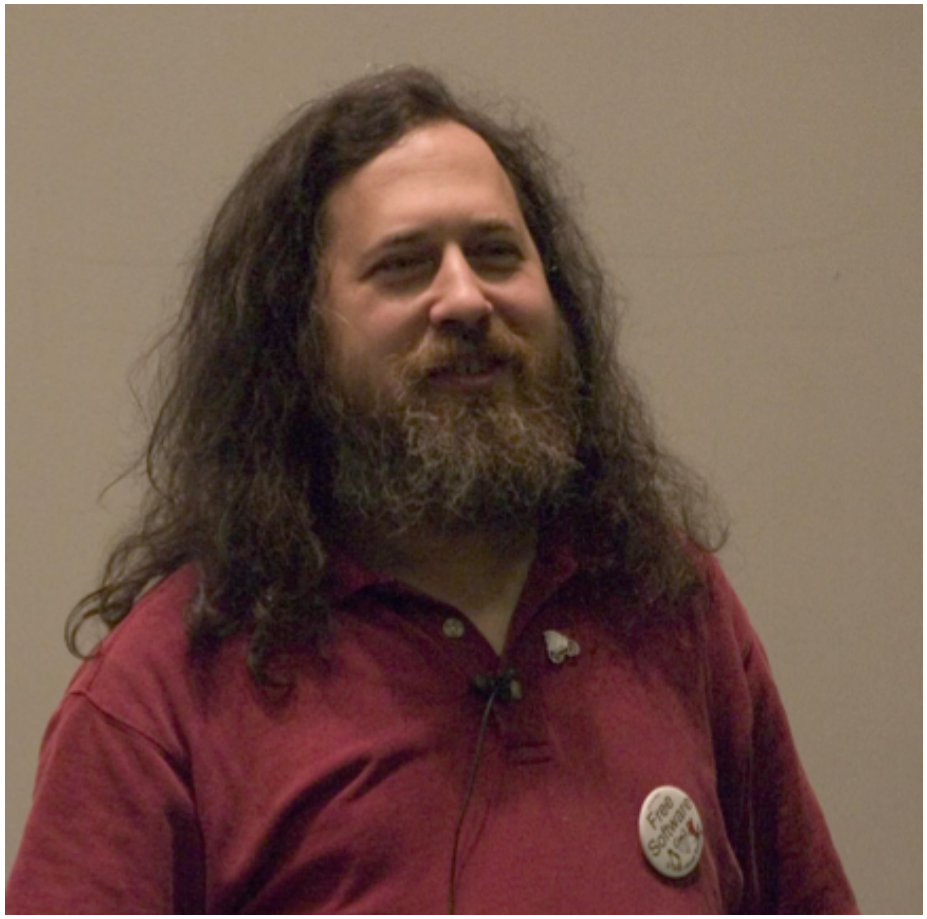
CLOUD A GOOD OR BAD IDEA ?

Steve Wozniak

I really worry about everything going to the cloud, I think it's going to be horrendous. I think there are going to be a lot of horrible problems in the next five years.

A lot of people feel, 'Oh, everything is really on my computer,' but I say the more we transfer everything onto the web, onto the cloud, the less we're going to have control over it.
(August 2012)

CLOUD A GOOD OR BAD IDEA ?



Richard Stallman

He said that cloud computing was simply a trap aimed at forcing more people to buy into locked, proprietary systems that would cost them more and more over time.

It's stupidity. It's worse than stupidity: it's a marketing hype campaign,

Somebody is saying this is inevitable – and whenever you hear somebody saying that, it's very likely to be a set of businesses campaigning to make it true.

(September 2008)

CLOUD A GOOD OR BAD IDEA ?

- Cloud computing enables companies and applications, which are system infrastructure dependent, to be **infrastructure-less**.
- By using the Cloud infrastructure on “pay as used and on demand”, all of us can **save in capital and operational investment**!
- Clients can:
  - Put their data on the platform instead of on their own desktop PCs and/or on their own servers.
  - They can put their applications on the cloud and use the servers within the cloud to do processing and data manipulations etc.

@Mark Baker

Cloud are transparent to users and applications, they can be built in multiple ways

branded products, proprietary open source, hardware or software, or just off-the-shelf PCs.

In general, they are built on **clusters of PC servers** and off-the-shelf components plus Open Source software combined with in-house applications and/or system software.

CHAPTER 2
DATA CENTER - INFRASTRUCTURE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LIBRARY TO THE WEB

SERVERS

1U

2U

4U

Source : dell.com

RACK

Source : server-racks.com

(POWER DISTRIBUTION UNIT, PDU)

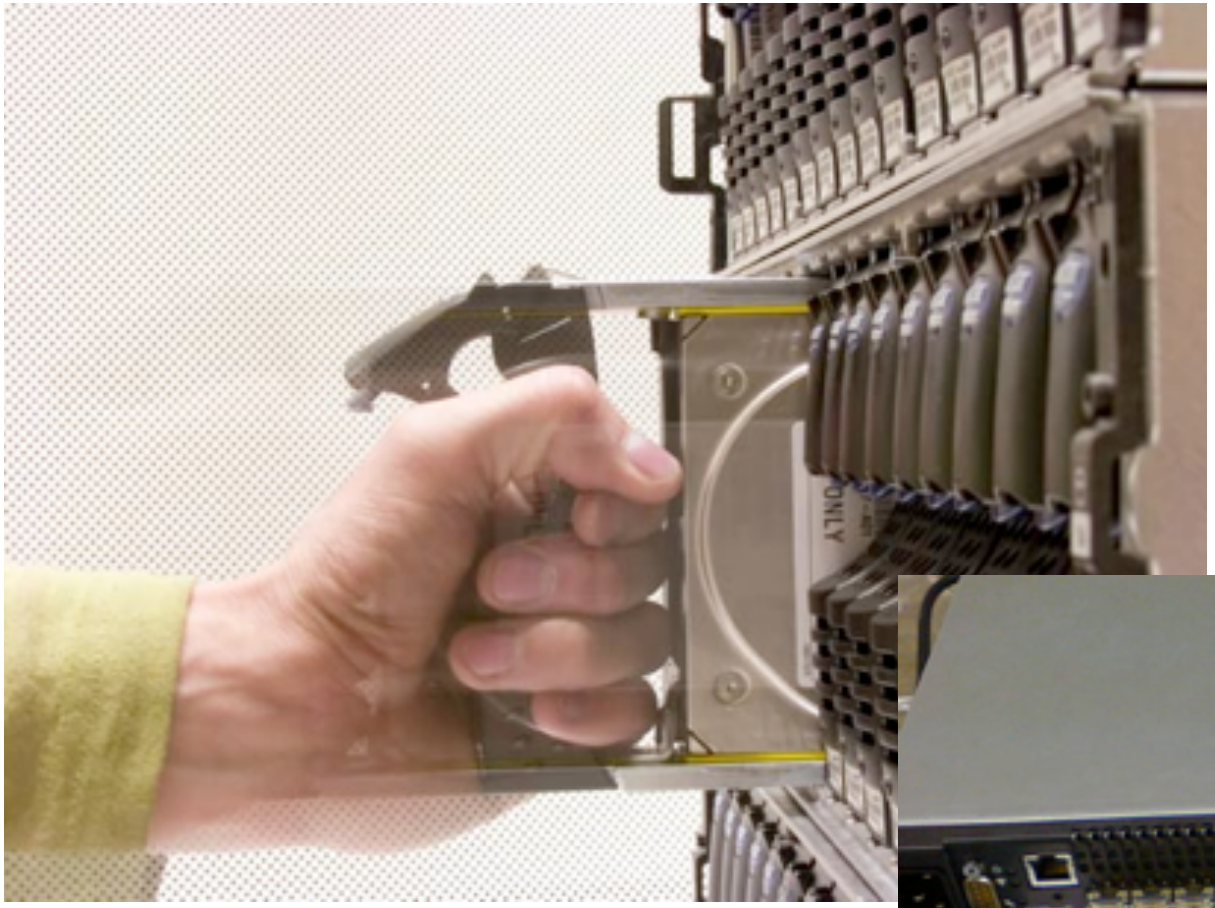
Source : 42u.com

Source : cloudtp.com

Switch

Source : webpage.pace.edu

STORAGE AREA NETWORK (SAN)

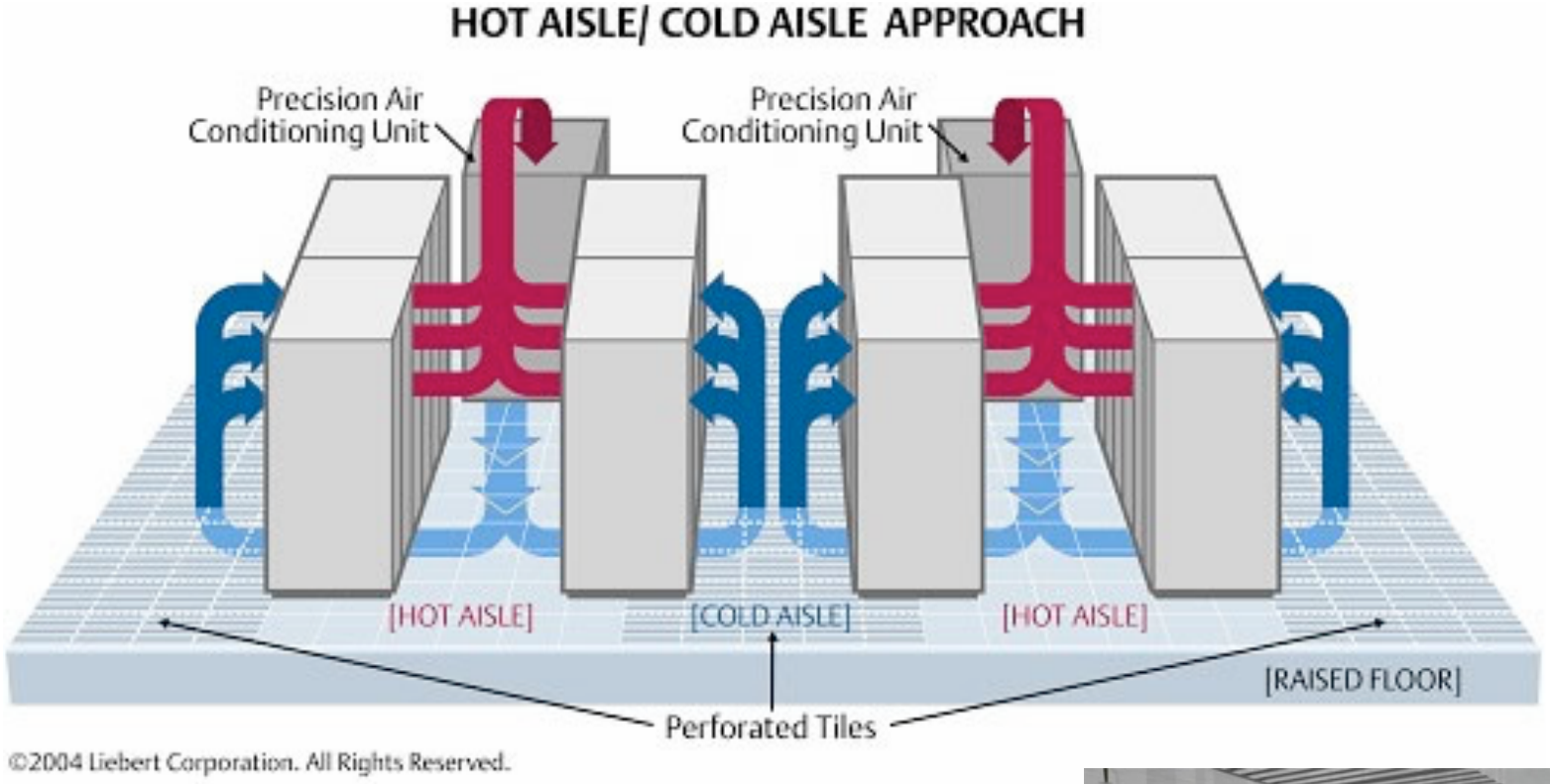


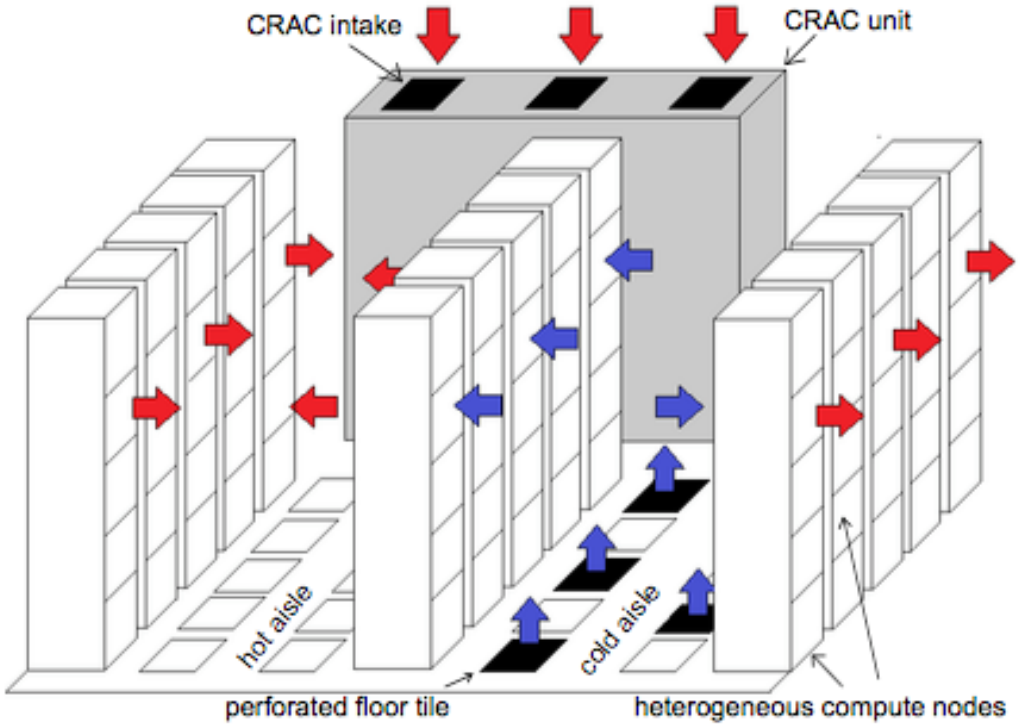
Source : businesscomputingworld.co.uk

Source : commons.wikimedia.org

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

COOLING


HOT AISLE/ COLD AISLE APPROACH
Precision Air Conditioning Unit
Precision Air Conditioning Unit
[HOT AISLE]
[COLD AISLE]
[HOT AISLE]
[RAISED FLOOR]
Perforated Tiles
©2004 Liebert Corporation. All Rights Reserved.


CRAC intake
CRAC unit
hot aisle
cold aisle
perforated floor tile
heterogeneous compute nodes

GOOGLE DATACENTER

DATA CENTER TIER LEVELS AND UPTIME

TIER

The concept of “Uptime” was pioneered by the Uptime Institute which was founded in 1993 and introduced its well defined Tier Classification system: I, II, III and IV, of which Tier IV represents the highest level of projected availability.

TIER 1 :
- Disponibility : 99.67%,
- no more 28.8 hours downtime/year
- No redundancy

TIER II :
- Disponibility : 99.75%,
- no more 22 hours downtime/year
- Partial redundancy

TIER III :
- Disponibility : 99.982 %,
- no more 1,5 hours downtime/year
- Redundancy N+1

TIER IV :
- Disponibility : 99.995 %,
- No more 0,8 hours downtime/year
- Redundancy 2N+1

http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

DATA CENTER TIER LEVELS AND UPTIME

TIER

The terms “N, N+1 and 2N”, typically refer to the number of power and cooling components that comprise the entire data center infrastructure systems.

Tier Requirement	Tier 1	Tier II	Tier III	Tier IV
Source	System	System	System	System + System
System Component Redundancy	N	N+1	N+1	Minimum of N+1
Distribution Paths	1	1	1 normal and 1 alternate	2 simultaneously active
Compartmentalization	No	No	No	Yes
Concurrently Maintainable	No	No	Yes	Yes
Fault Tolerance (single event)	No	No	No	Yes

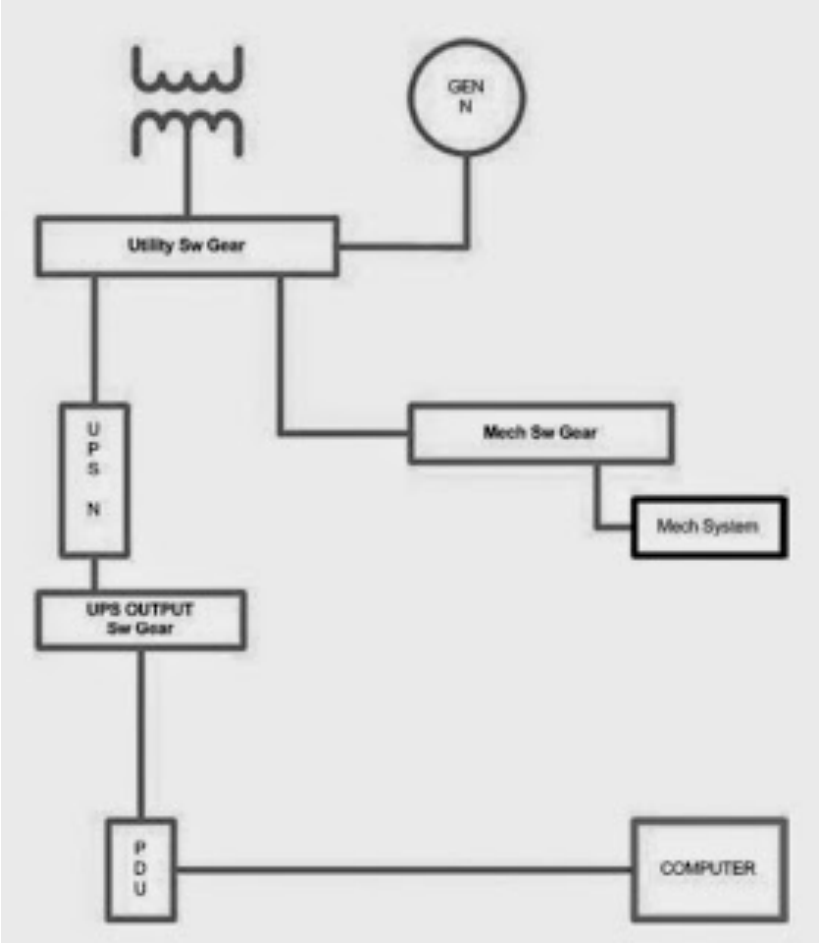
http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

DATA CENTER TIER LEVELS AND UPTIME

TIER I

TIER I :

- Components capacity to support IT load is 100%
- Single distribution path
- Rack power < 1kW Multi-points of failure and human errors
- Schedule maintenance twice per annum last for 12 hours
- Availability 99.67%, annual down time max. 28.8 hours



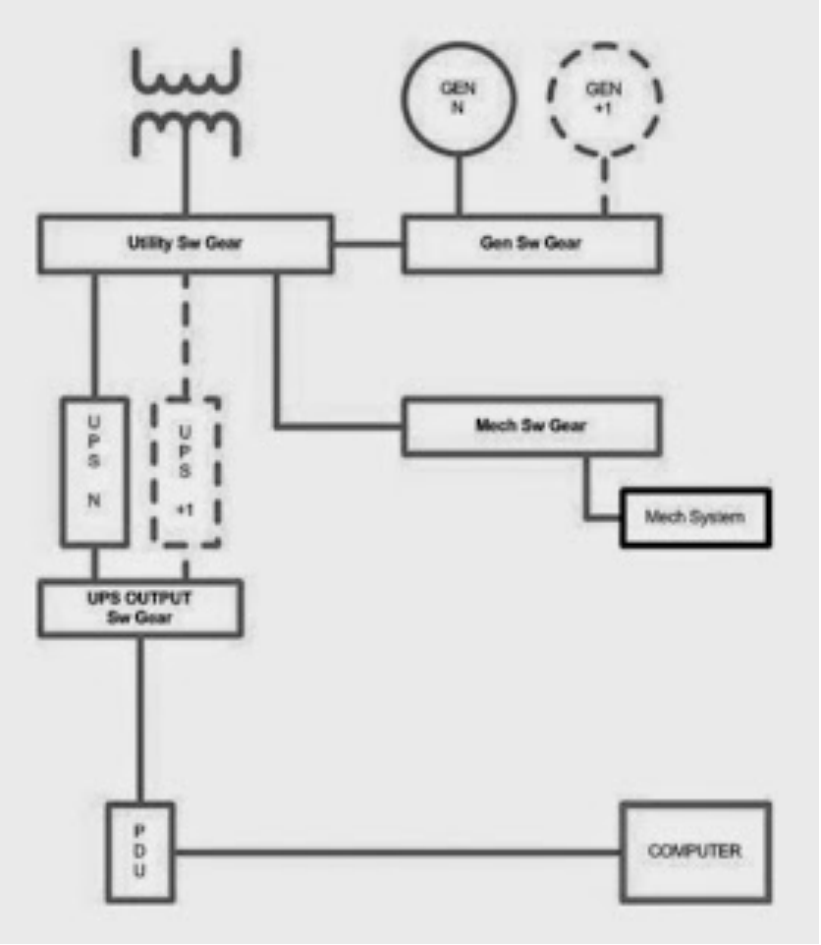
http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

DATA CENTER TIER LEVELS AND UPTIME

TIER II

TIER II :

- Components capacity to support IT load is N+1
- Redundant for critical components
- Single distribution path
- Rack power < 2kW
- Multi-points of failure and human errors
- Schedule maintenance 3 times every 2 years each 12 hours
- Availability 99.75%, annual down time max. 22 hours



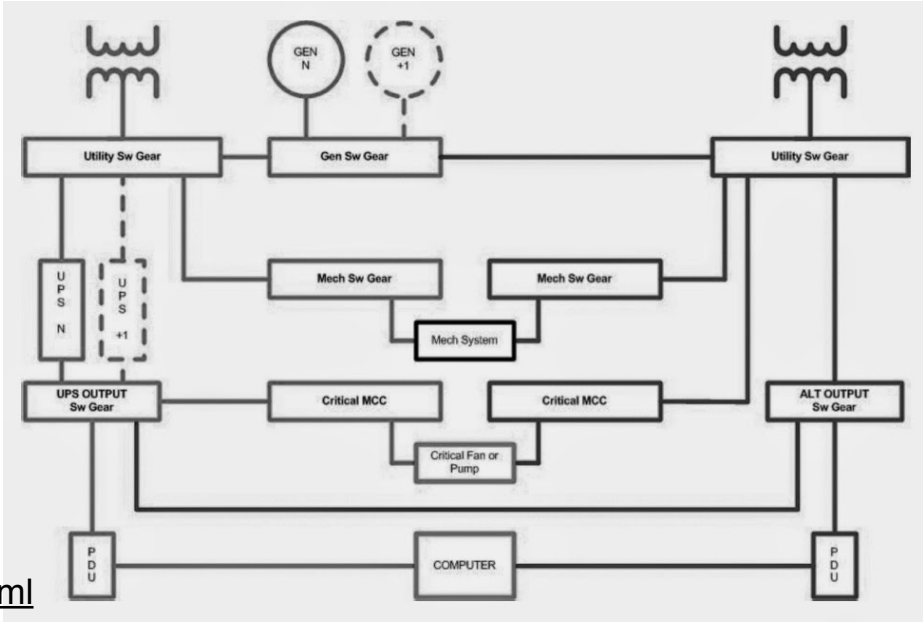
http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

DATA CENTER TIER LEVELS AND UPTIME

TIER III

TIER II :

- Components capacity to support IT load is N+1
- Redundant for all components and distribution path
- Allow concurrent maintenance
- Rack power > 3kW
- Require MV supply (11kV in Hong Kong)
- Some failure points and human errors
- Schedule maintenance not required
- Availability 99.98%, annual down time max. 96 mins



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

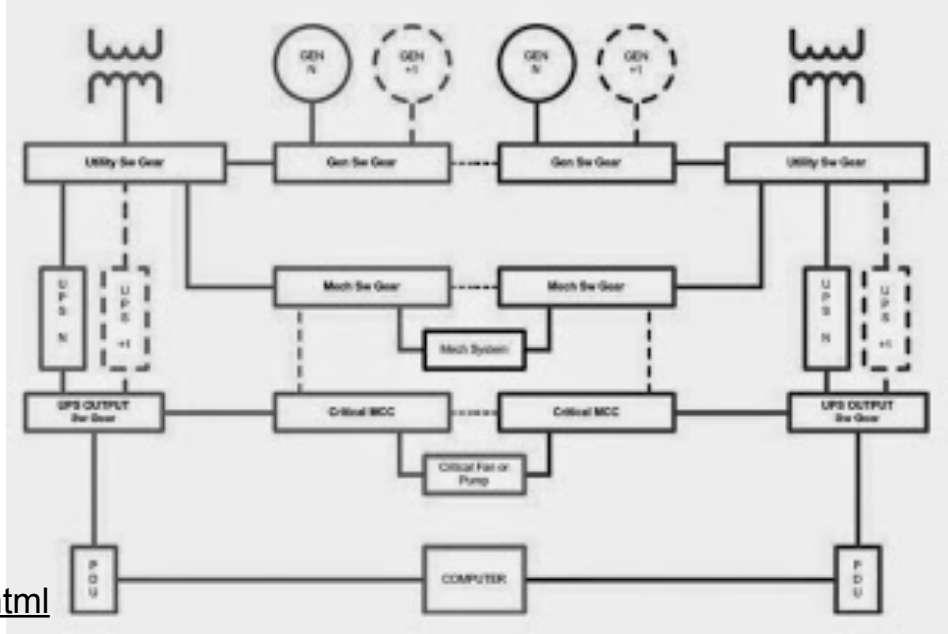
http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

DATA CENTER TIER LEVELS AND UPTIME

TIER IV

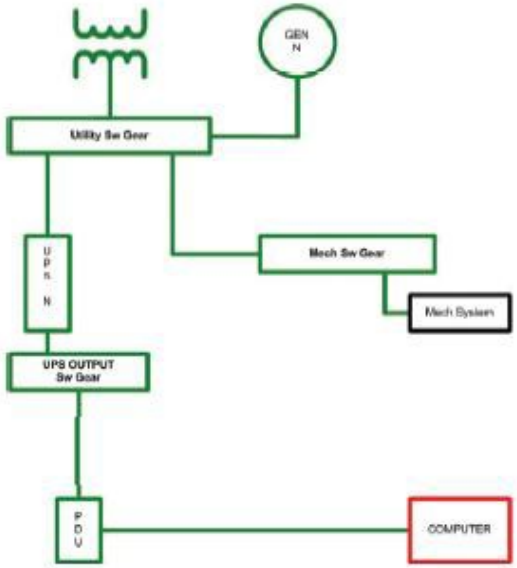
TIER IV :

- Components capacity to support IT load is N after any failure
- Dual active distribution path
- No Single Point of Failure
- Rack power > 4kW
- Require MV supply (11kV in Hong Kong)
- Fault Tolerant, Fire, EPO and human errors
- Schedule maintenance not required
- Availability 99.99%, annual down time max. 48 mins

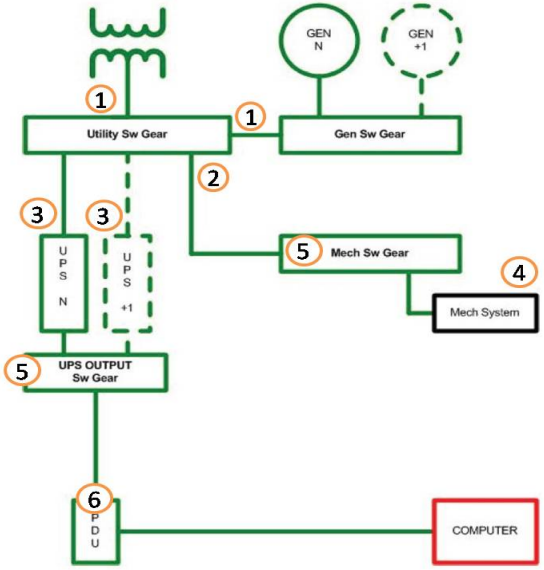


http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

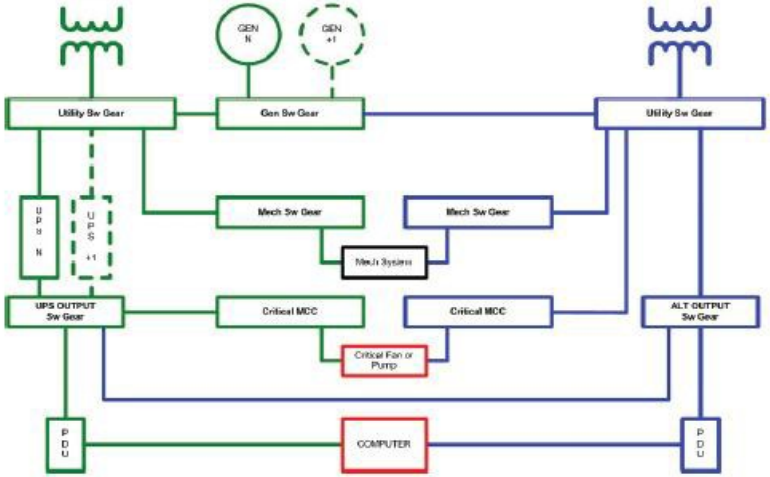
Tier I



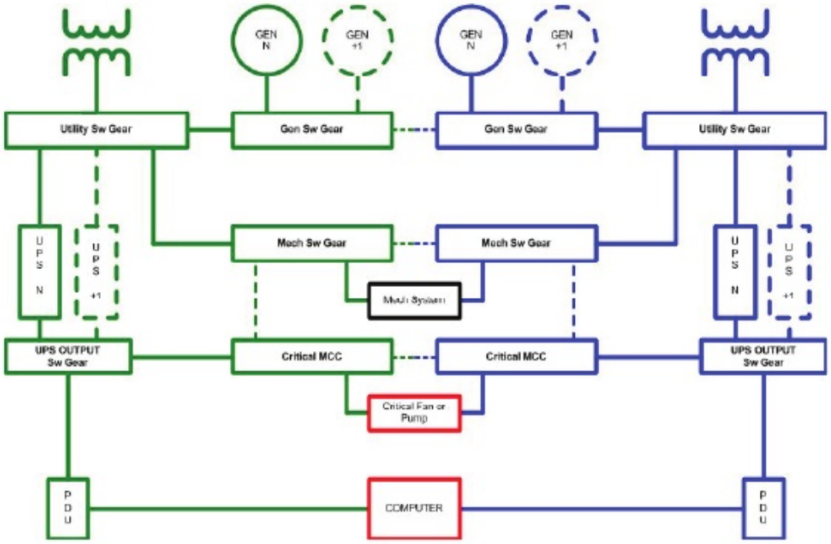
Tier II



Tier III



Tier IV

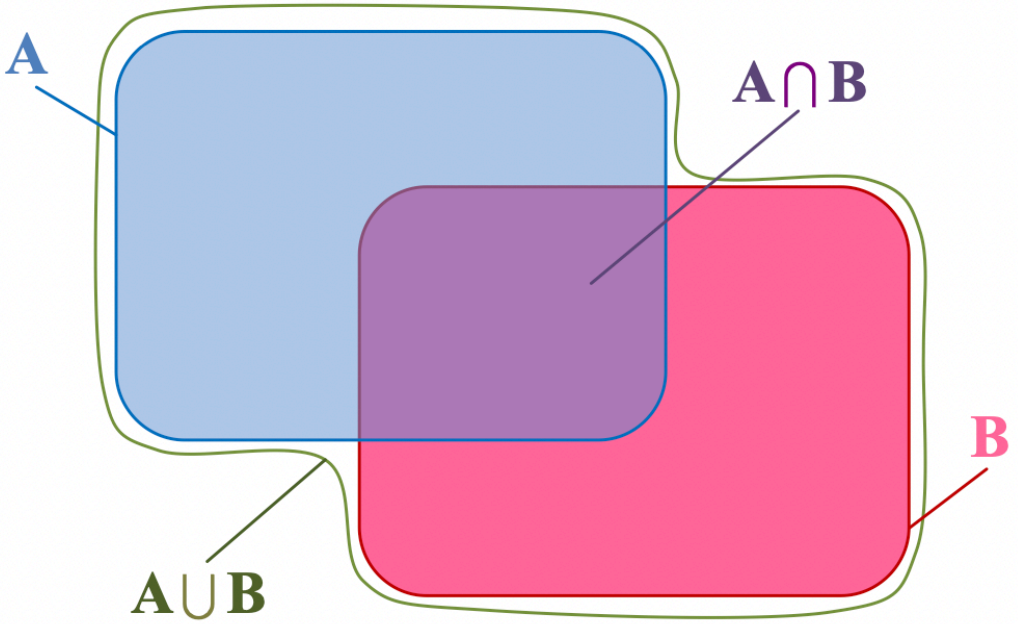


IMT Atlantiqu
Bretagne-Pays de
École Mines-Télécc

Définitions :

L'événement "A et B", noté A ∩ B, est réalisé lorsque les deux événements A et B sont simultanément réalisés.

L'événement "A ou B", noté A ∪ B, est réalisé lorsqu'au moins l'un des deux événements est réalisé.



Théorème :
Si A et B sont deux événements d'une expérience aléatoire, alors :
$P(\mathrm{A} \cup \mathrm{B}) = P(\mathrm{A}) + P(\mathrm{B}) - P(\mathrm{A} \cap \mathrm{B})$ $P(\mathrm{A} \cap \mathrm{B})$ = P(A)xP(B)

Définition :
On dit que deux événements A et B sont incompatibles si $\mathrm{A} \cap \mathrm{B} = \varnothing$.

Propriété :
Si deux événements A et B sont incompatibles alors $P(\mathrm{A} \cup \mathrm{B}) = P(\mathrm{A}) + P(\mathrm{B})$.

DATA CENTER TIER LEVELS AND UPTIME

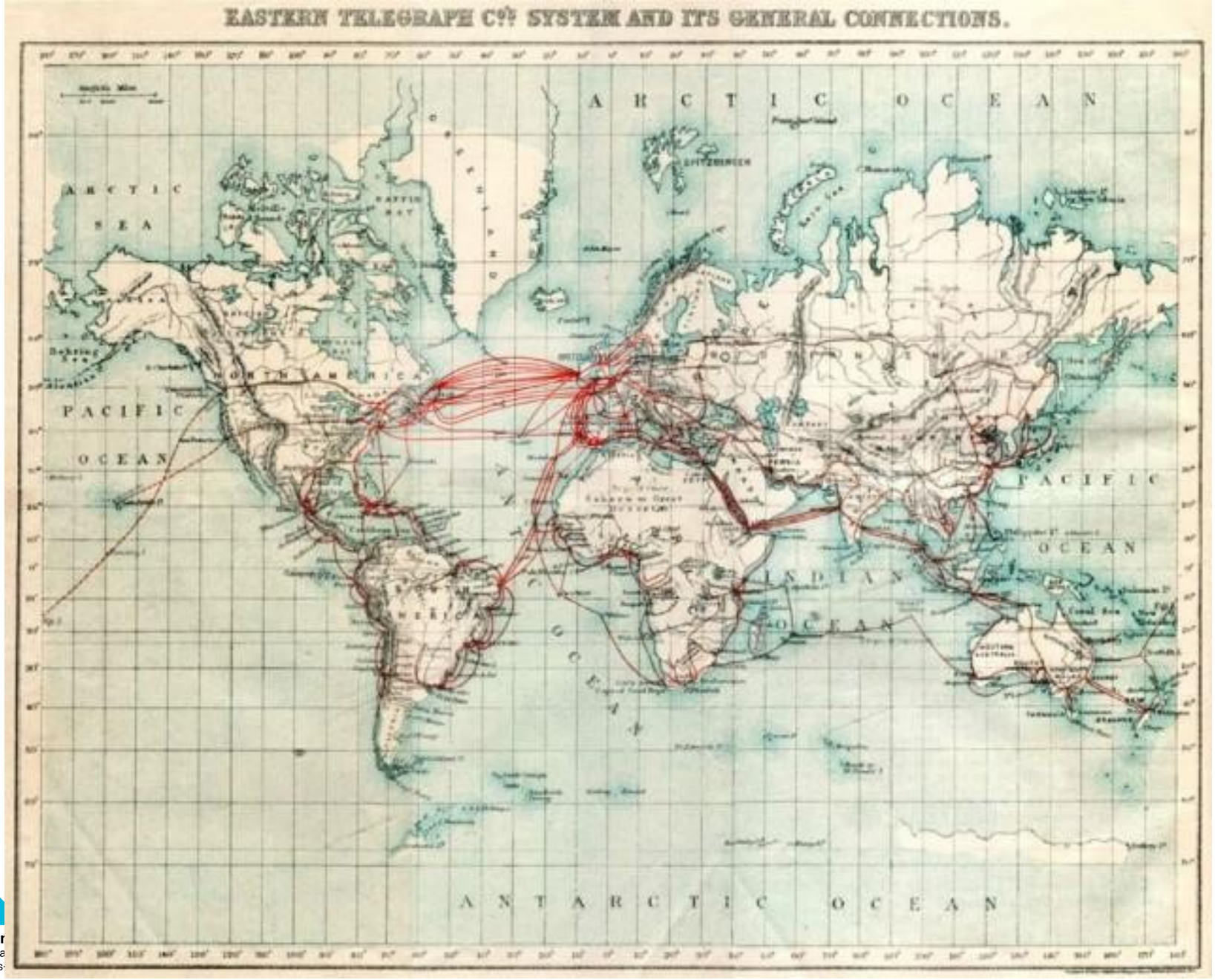
An “N” system is not redundant at all, and the failure of any component will cause an outage, effectively describing a tier 1 type facility.

N+1 and 2N, represent increasing levels of component redundancies and power paths, roughly mapping to the tiers 2-4, however it is **important to understand that redundant components in themselves do not guarantee continuous availability.**

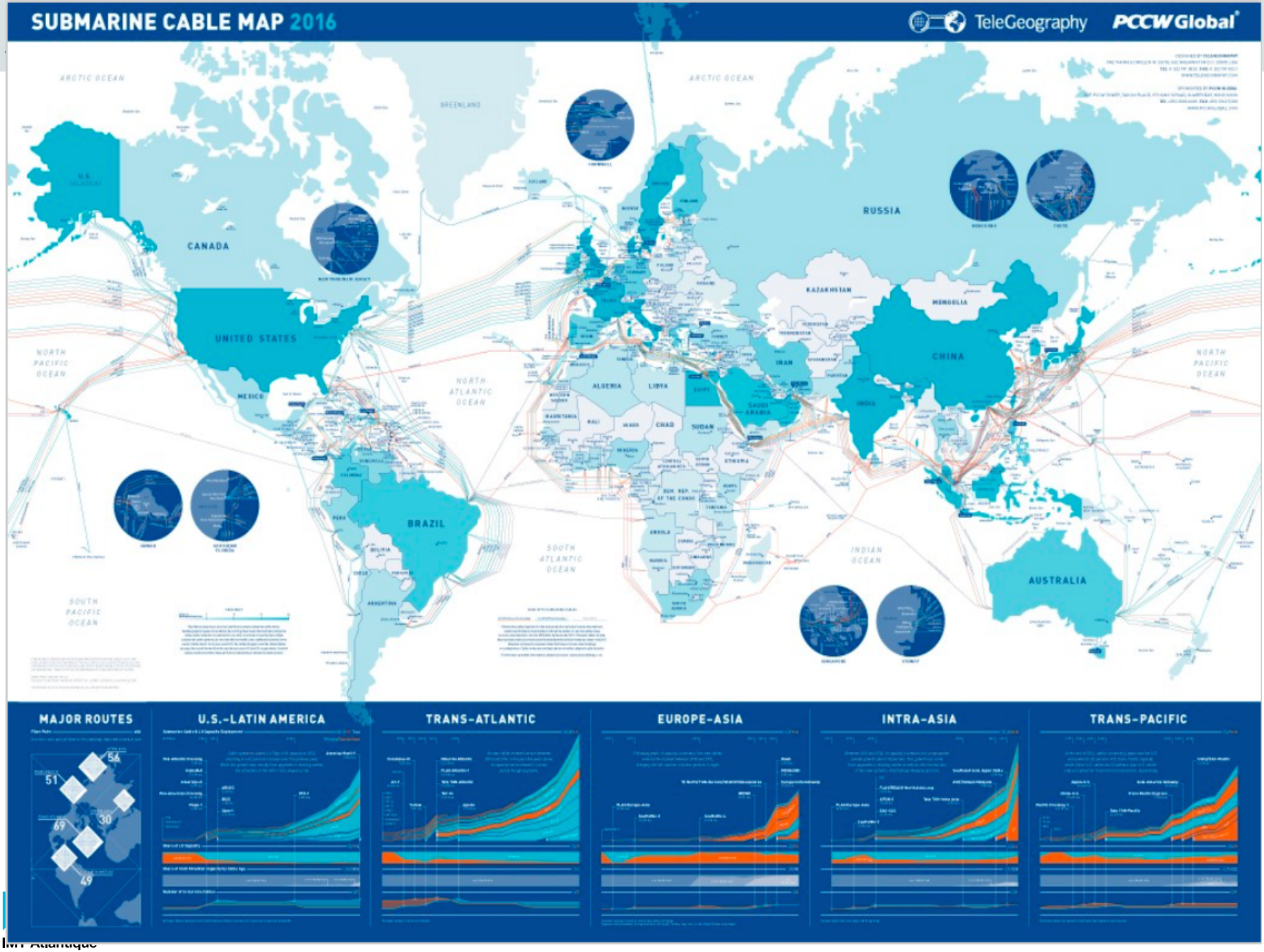
Besides redundancy, the ability to do planned maintenance or emergency repairs on systems may involve the necessity to take them offline

http://green-data.blogspot.fr/2014/09/more-about-data-center-tier-levels.html

TELEGRAPH 1901

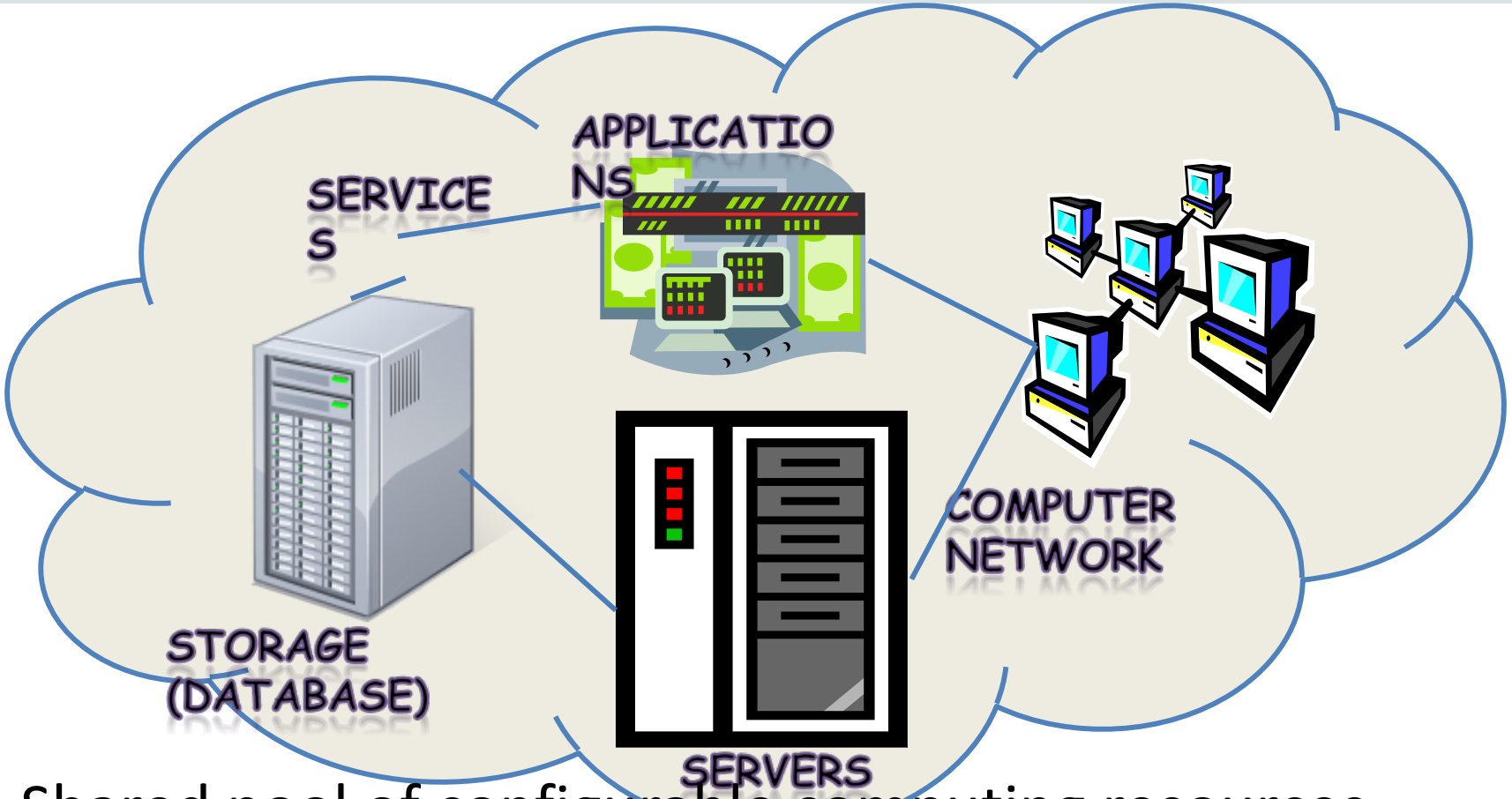


WORLD NETWORK



CHAPTER 3
INSIDE THE CLOUD MODEL

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



- Shared pool of configurable computing resources
- On-demand network access
- Provisioned by the Service Provider

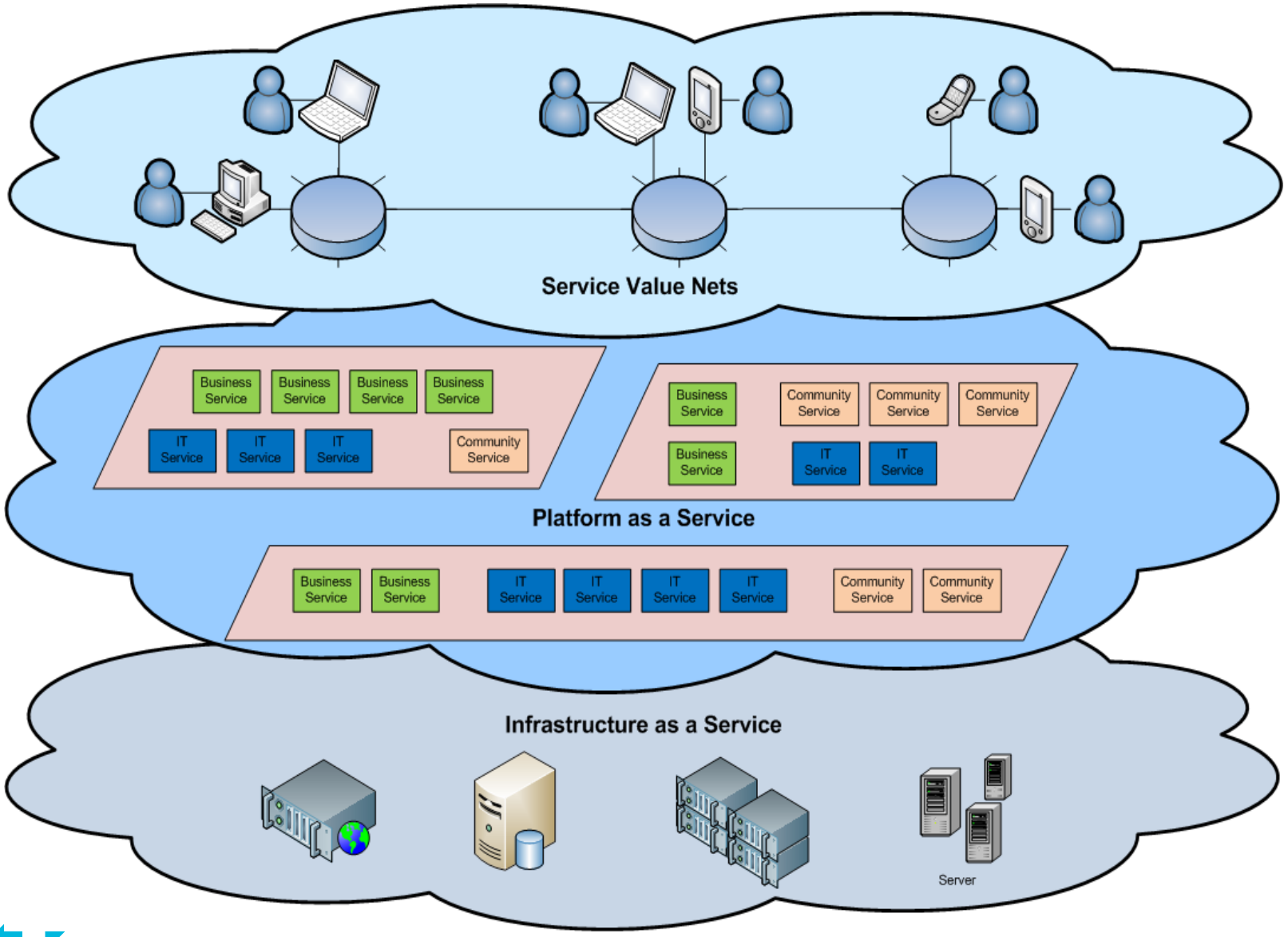
CLOUD COMPUTING CHARACTERISTICS

Common Characteristics:

- Massive Scale
- Resilient Computing
- Homogeneity
- Geographic Distribution
- Virtualization
- Service Orientation
- Low Cost Software
- Advanced Security

Essential Characteristics:

- On Demand Self-Service
- Broad Network Access
- Rapid Elasticity
- Resource Pooling
- Measured Service


Service Value Nets
Business Service
Business Service
Business Service
Business Service
IT Service
IT Service
IT Service
Community Service
Business Service
Community Service
Community Service
Community Service
Business Service
IT Service
IT Service
Platform as a Service
Business Service
Business Service
IT Service
IT Service
IT Service
IT Service
Community Service
Community Service
Infrastructure as a Service
Server

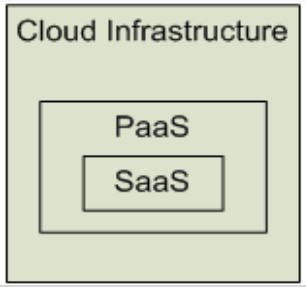
@Mark Baker

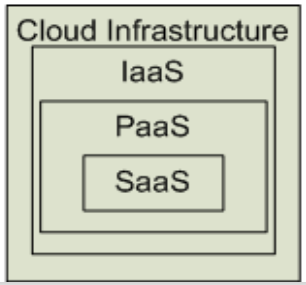
CLOUD SERVICE MODELS

Software as a Service (SaaS)	Platform as a Service (PaaS)	Infrastructure as a Service (IaaS)

SalesForce CRM

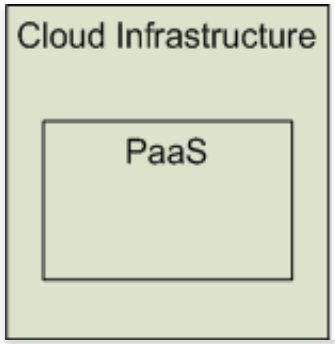
LotusLive

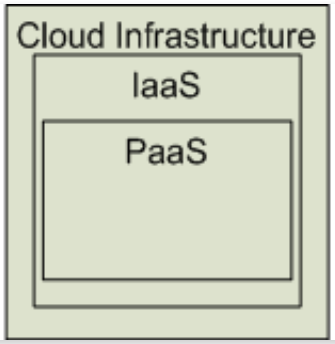




Software as a Service (SaaS) Providers Applications

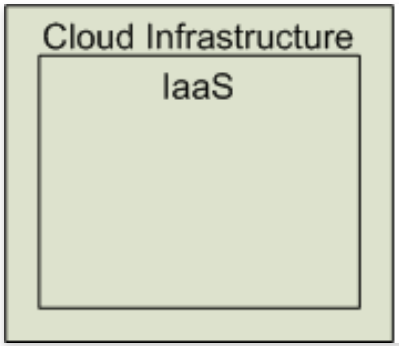
Google App Engine





Platform as a Service (PaaS)

Deploy customer created Applications



Infrastructure as a Service (IaaS)

Rent Processing, storage, N/W capacity & computing resources

Application Service (SaaS)	MS Live/ExchangeLabs, IBM, Google Apps; Salesforce.com Quicken Online, Zoho, Cisco
Application Platform	Google App Engine, Mosso, Force.com, Engine Yard, Facebook, Heroku, AWS
Server Platform	3Tera, EC2, SliceHost, GoGrid, RightScale, Linode
Storage Platform	Amazon S3, Dell, Apple, ...

@Mark Baker

	Services	Description
Application Focused	Services	Services – Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
	Application	Application – Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
	Development	Development – Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as SalesForce
Infrastructure Focused	Platform	Platform – Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
	Storage	Storage – Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
	Hosting	Hosting – Physical data centers such as those run by IBM, HP, NaviSite, etc.

CHAPTER 4
IAS
INFRASTRUCTURE AS A SERVICE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CHAPTER 4.1
VIRTUALIZATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Aggregation

– Make N resources appear as 1 (clusters / grids)

Partition / Replication:

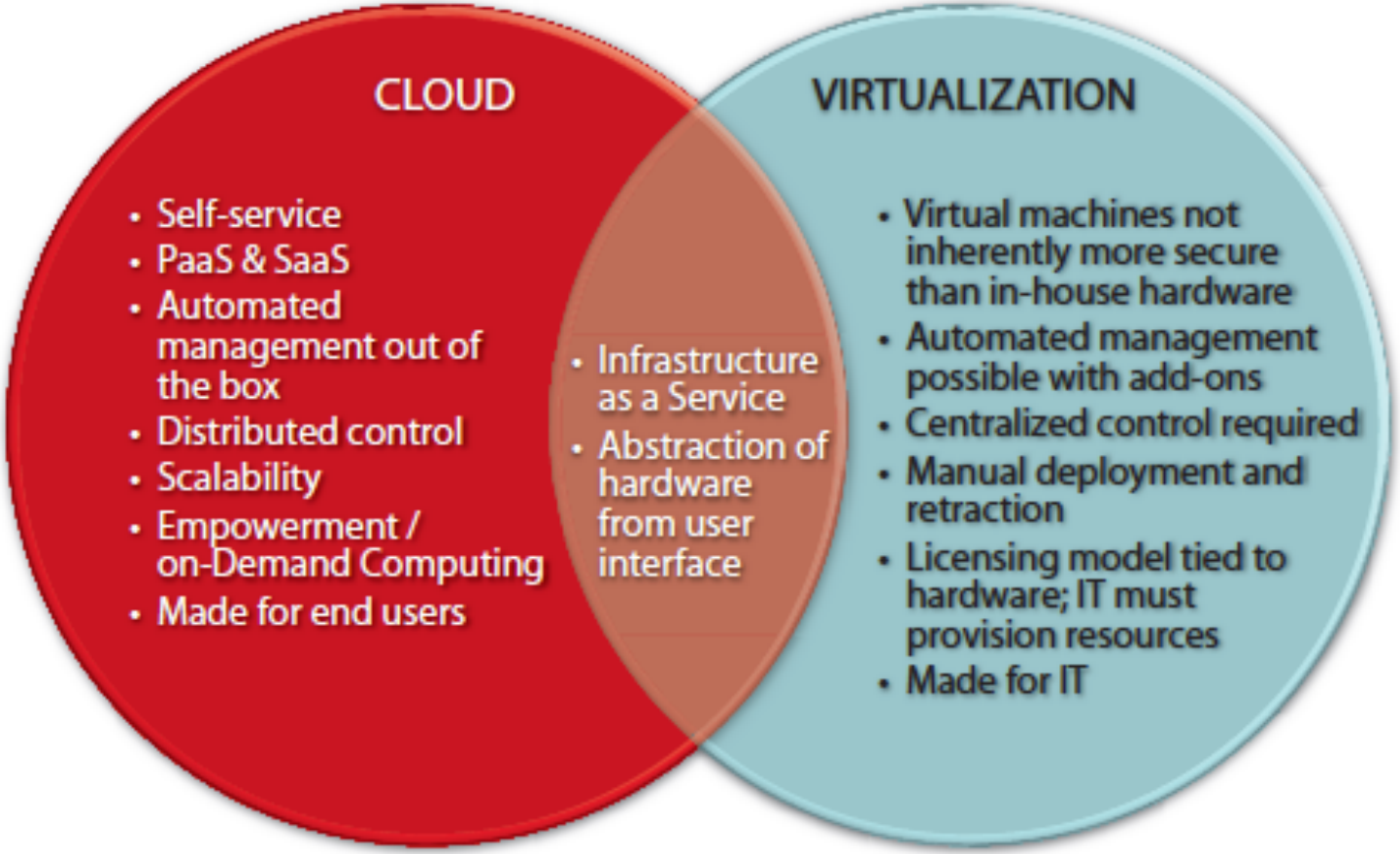
– Make a resource appear as N

Translation (Emulation)

– Make X appear as Y (sometimes X is identical to Y)

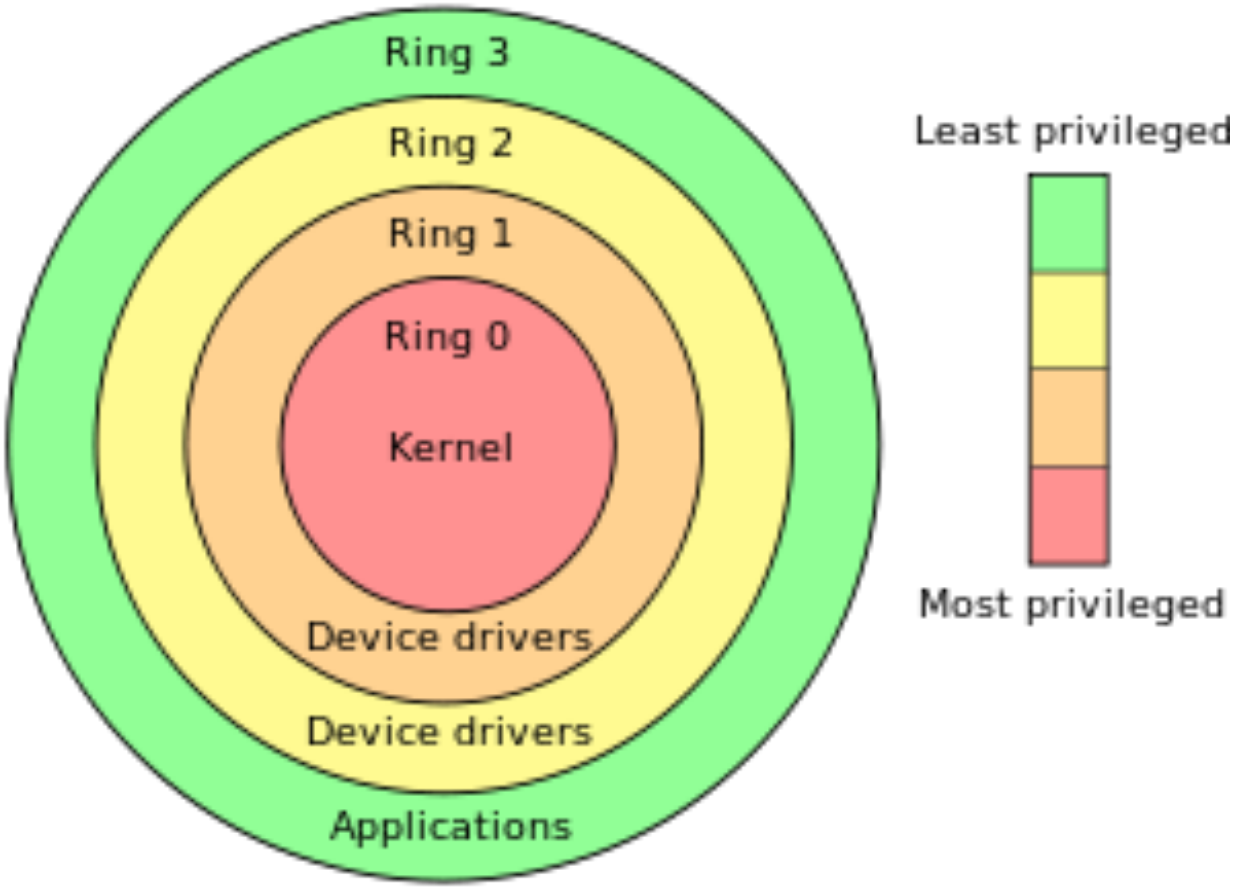
– May be combined with “partition”

Mostly interested in “partition”


CLOUD
• Self-service
• PaaS & SaaS
• Automated management out of the box
• Distributed control
• Scalability
• Empowerment / on-Demand Computing
• Made for end users
• Infrastructure as a Service
• Abstraction of hardware from user interface
VIRTUALIZATION
• Virtual machines not inherently more secure than in-house hardware
• Automated management possible with add-ons
• Centralized control required
• Manual deployment and retraction
• Licensing model tied to hardware; IT must provision resources
• Made for IT

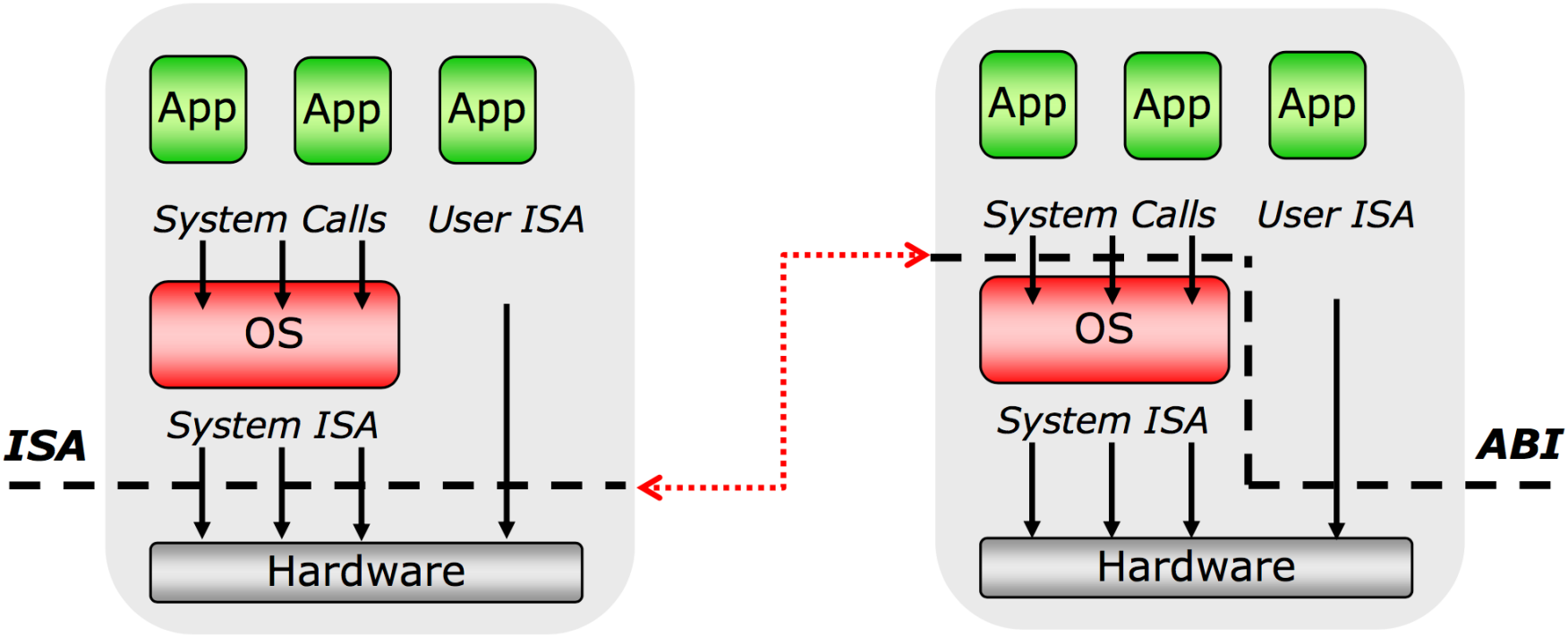
- First appeared in IBM mainframes in 1972
- Allowed multiple users to share a batch-oriented system
- Formal definition of virtualization helped move it beyond IBM
  - A VMM provides an environment for programs that is essentially identical to the original machine
  - Programs running within that environment show only minor performance decreases
  - The VMM is in complete control of system resources
- In late 1990s Intel CPUs fast enough for researchers to try virtualizing on general purpose PCs
  - Xen and VMware created technologies, still used today
  - Virtualization has expanded to many OSes, CPUs, VMMs

INSIDE VIRTUALISATION



VIRTUAL MACHINE INTERFACES

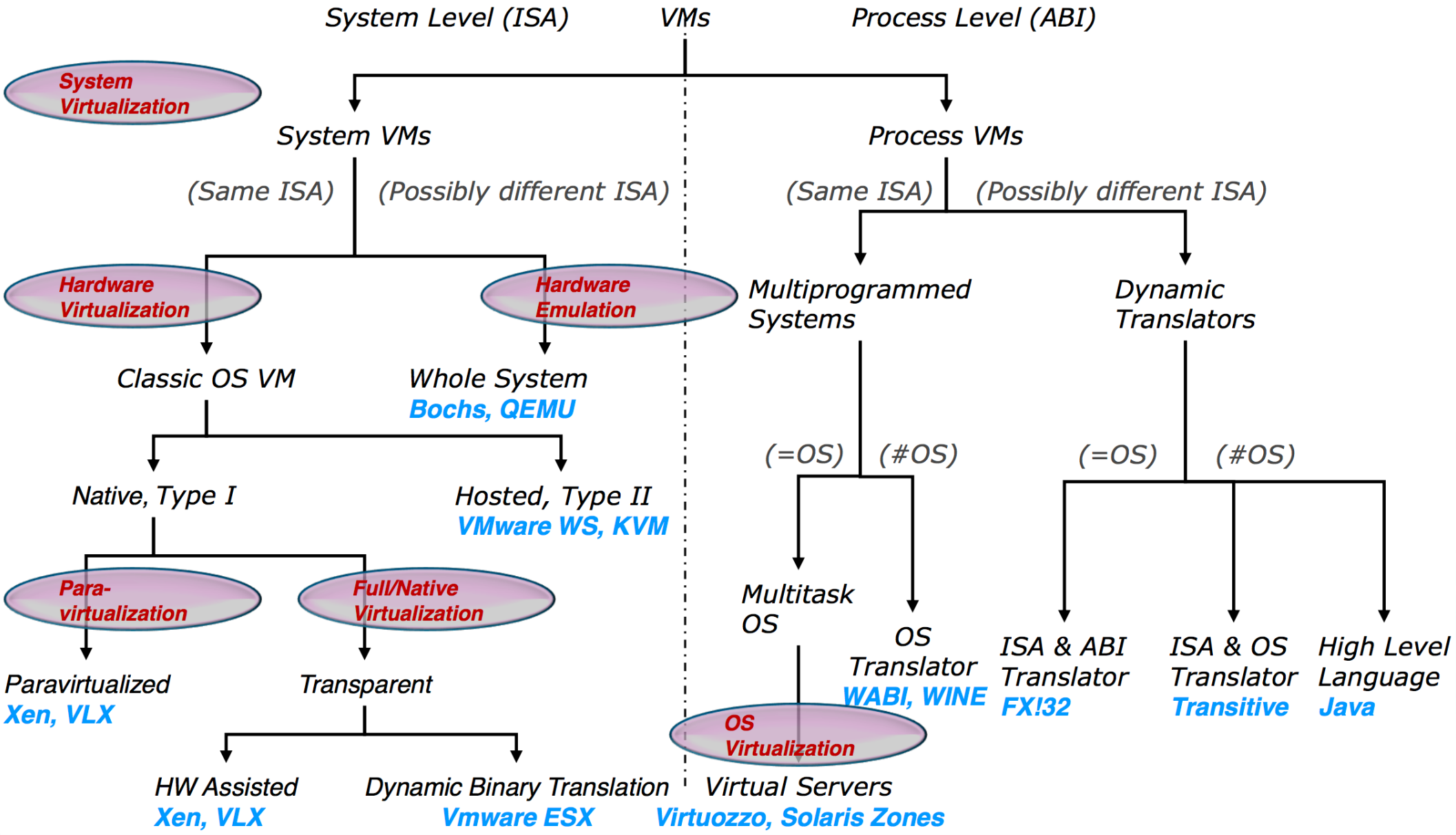
- Systems built of : hardware, OS, applications
- 2 main interfaces: ISA (hardware), ABI (OS)



System level VM provide an ISA interface

Process level VM provide an ABI interface

TAXONOMY (DERIVED FROM E. SMITH & NAIR)

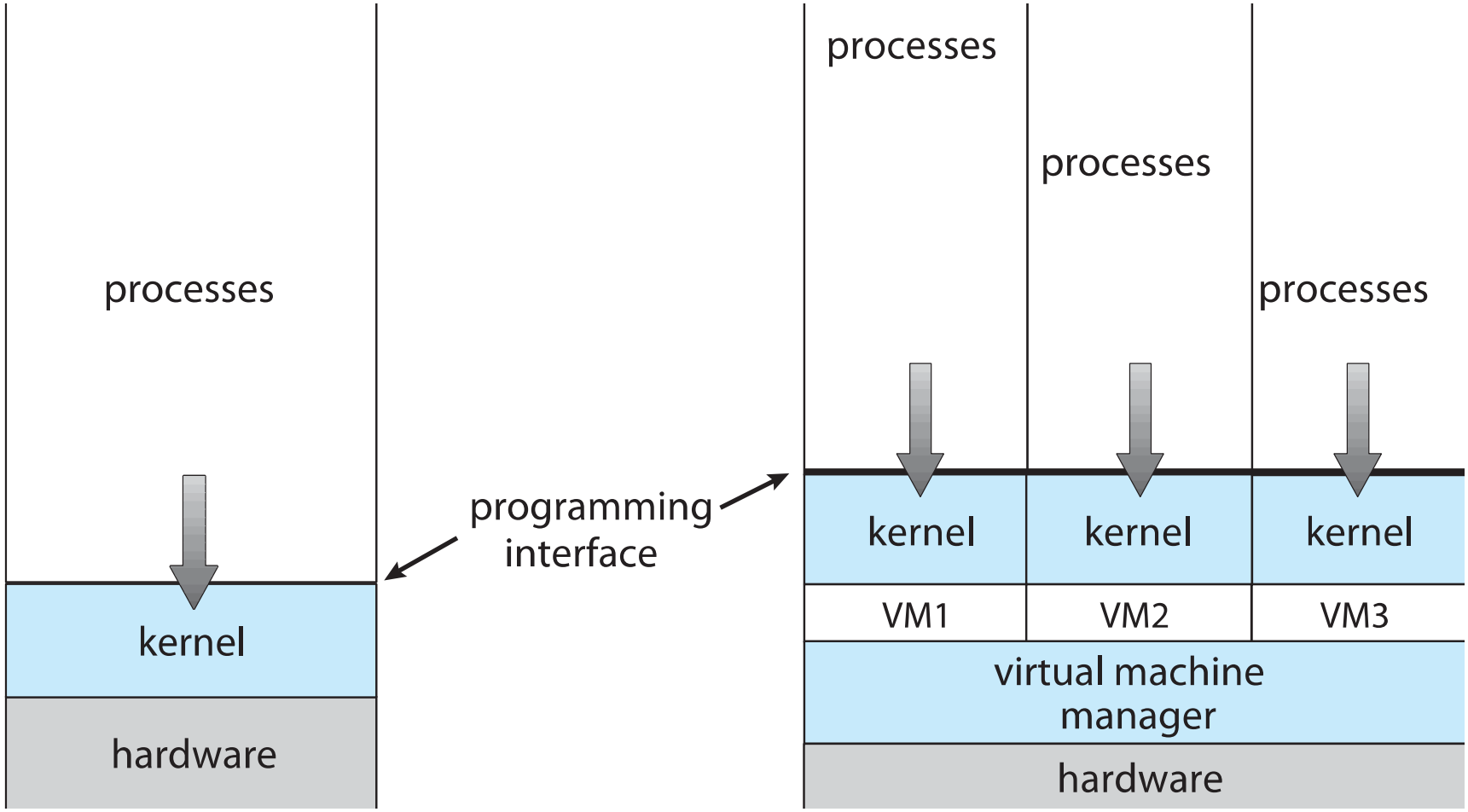


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

OVERVIEW OF VIRTUAL MACHINE CONCEPT

- Fundamental idea – abstract hardware of a single computer into several different execution environments
  - Similar to layered approach
  - But layer creates virtual system (**virtual machine**, or **VM**) on which operation systems or applications can run
- Several components
  - **Host** – underlying hardware system
  - **Virtual machine manager** (**VMM**) or **hypervisor** – creates and runs virtual machines by providing interface that is ***identical*** to the host
    - (Except in the case of paravirtualization)
  - **Guest** – process provided with virtual copy of the host
    - Usually an operating system
- Single physical machine can run multiple operating systems concurrently, each in its own virtual machine

SYSTEM MODEL



(a) Nonvirtual machine

(b) Virtual machine

IMPLEMENTATION OF VMMS

- Vary greatly, with options including:
  - **Type 0 hypervisors** - Hardware-based solutions that provide support for virtual machine creation and management via firmware
    - IBM LPARs and Oracle LDOMs are examples
  - **Type 1 hypervisors** - Operating-system-like software built to provide virtualization
    - Including VMware ESX, Joyent SmartOS, and Citrix XenServer
  - **Type 1 hypervisors** – Also includes general-purpose operating systems that provide standard functions as well as VMM functions
    - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM
  - **Type 2 hypervisors** - Applications that run on standard operating systems but provide VMM features to guest operating systems
    - Including VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox

CLASSIC OS VMS

Issues:

- Run multiple OS's
- OS designed assuming it is the only software controlling the physical resources of the machine.
- Need to detect and resolve conflicts such as masking interrupts, initiating an I/O, or programming the MMU by providing the expected behavior within the Virtual Machine, not necessarily on the physical level.
- In brief: detect sensitive instructions and fake them.
  - User mode to trap such instruction upon execution
  - Modify OS ahead of time
  - Hardware support

- Goal:
  - Run the binary guest OS in user mode though it's been written to run in supervisor mode.
- Means:
  - Transparent Virtualization: (full or native)
    - No modification of the OS image
    - Fully Virtualizable Processors (VT-x, AMD-V, IBM PPC)
    - Dynamic Binary Translation (VMware)
  - Para-virtualization:
    - Modification of some of the OS HAL source files
    - (can be seen as a port to a new processor very similar to the real one).

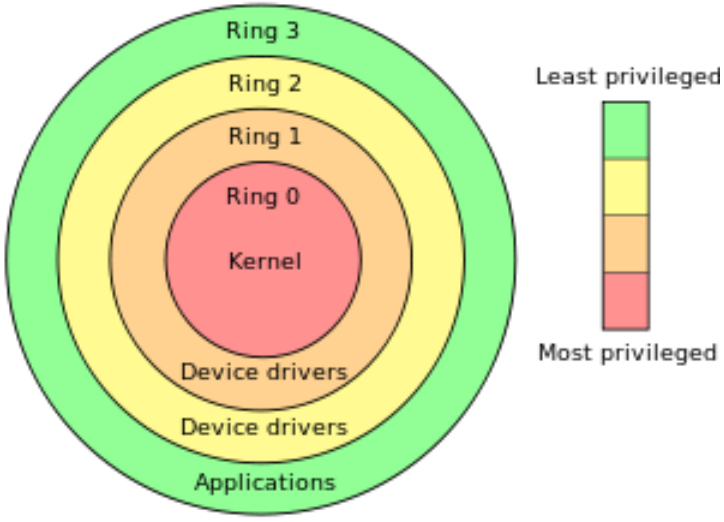
IMPLEMENTATION OF VMMS (CONT.)

- Other variations include:
  - **Paravirtualization** - Technique in which the guest operating system is modified to work in cooperation with the VMM to optimize performance
  - **Programming-environment virtualization** - VMMs do not virtualize real hardware but instead create an optimized virtual system
    - Used by Oracle Java and Microsoft.Net
  - **Emulators** – Allow applications written for one hardware environment to run on a very different hardware environment, such as a different type of CPU
  - **Application containment** - Not virtualization at all but rather provides virtualization-like features by segregating applications from the operating system, making them more secure, manageable
    - Including Oracle Solaris Zones, BSD Jails, and IBM AIX WPARs
- Much variation due to breadth, depth and importance of virtualization in modern computing

https://instances.vantage.sh

BUILDING BLOCKS

- Generally difficult to provide an ***exact*** duplicate of underlying machine
  - Especially if only dual-mode operation available on CPU
  - But getting easier over time as CPU features and support for VMM improves
  - Most VMMs implement **virtual CPU** (**VCPU**) to represent state of CPU per guest as guest believes it to be
    - When guest context switched onto CPU by VMM, information from VCPU loaded and stored
  - Several techniques, as described in next slides

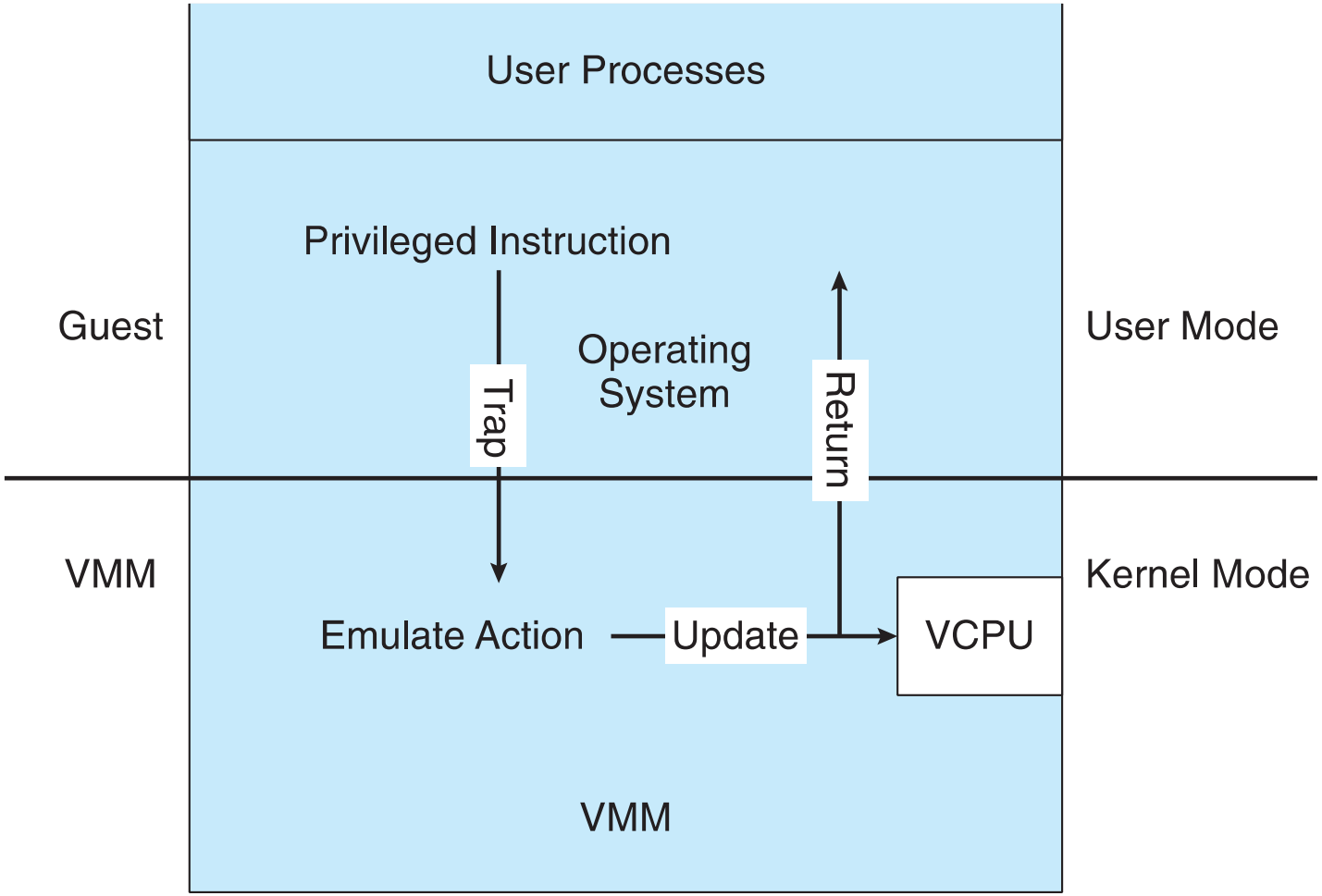


BUILDING BLOCKS

- Dual mode CPU means guest executes in user mode
  - Kernel runs in kernel mode
  - Not safe to let guest kernel run in kernel mode too
  - So VM needs two modes – virtual user mode and virtual kernel mode
    - Both of which run in real user mode
  - Actions in guest that usually cause switch to kernel mode must cause switch to virtual kernel mode

- How does switch from virtual user mode to virtual kernel mode occur?
  - Attempting a privileged instruction in user mode causes an error -> trap
  - VMM gains control, analyzes error, executes operation as attempted by guest
  - Returns control to guest in user mode
  - Known as **trap-and-emulate**
  - Most virtualization products use this at least in part
- Trap to supervisor mode when executed from user mode
  - cli, sti (Intel x86) trap when executed from user-mode control to guest in user mode

BUILDING BLOCKS - TRAP AND EMULATE



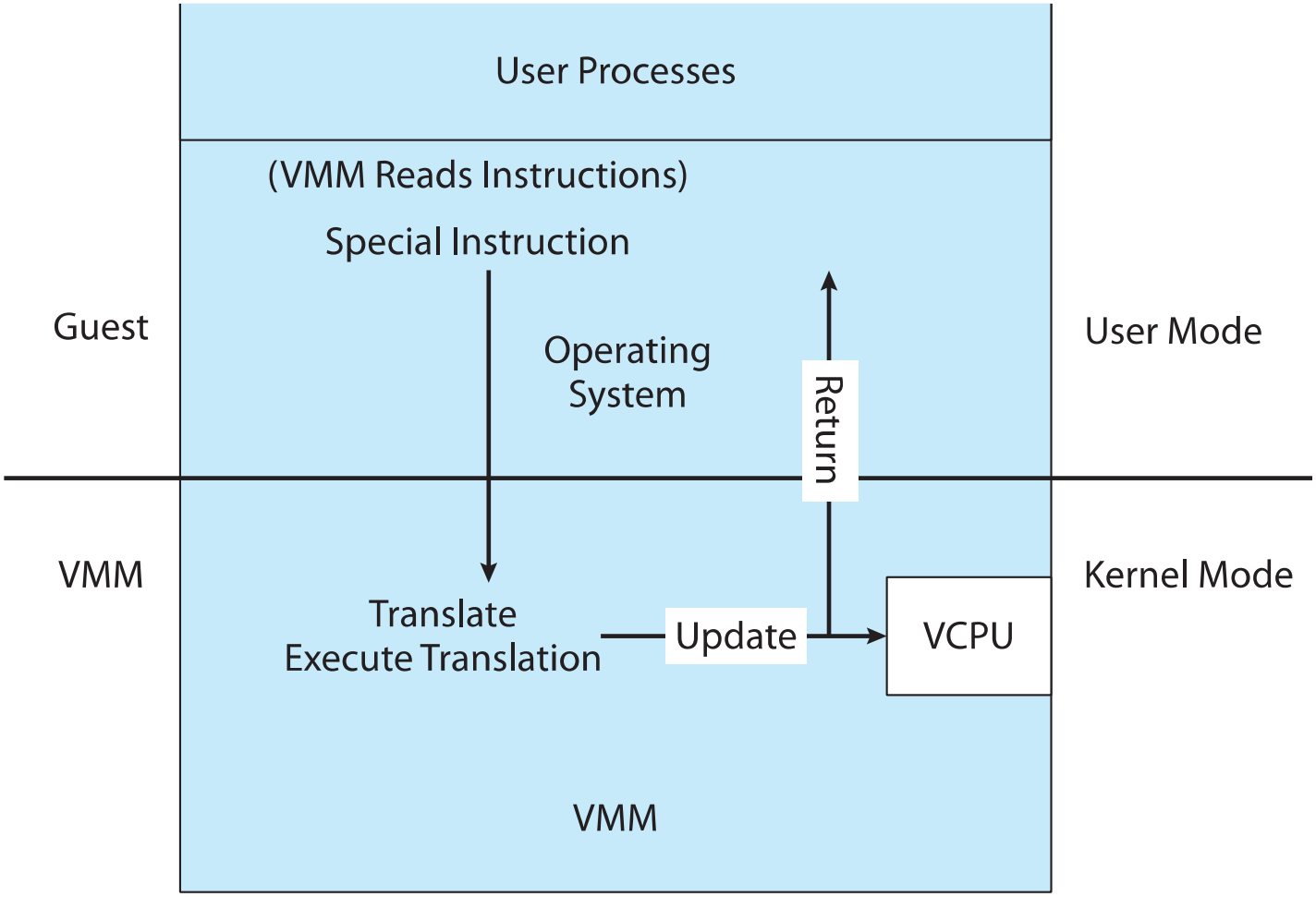
- User mode code in guest runs at same speed as if not a guest
- But kernel mode privilege mode code runs slower due to trap-and-emulate
  - Especially a problem when multiple guests running, each needing trap-and-emulate
- CPUs adding hardware support, mode CPU modes to improve virtualization performance

- Some CPUs don't have clean separation between privileged and nonprivileged instructions
  - Earlier Intel x86 CPUs are among them
    - Earliest Intel CPU designed for a calculator
  - Backward compatibility means difficult to improve
  - Consider Intel x86 `popf` instruction
    - Loads CPU flags register from contents of the stack
    - If CPU in privileged mode -> all flags replaced
    - If CPU in user mode -> on some flags replaced
      - No trap is generated

BUILDING BLOCKS - BINARY TRANSLATION

- Other similar problem instructions we will call ***special instructions***
  - Caused trap-and-emulate method considered impossible until 1998
- Binary translation solves the problem
  - Basics are simple, but implementation very complex
  1. If guest VCPU is in user mode, guest can run instructions natively
  2. If guest VCPU in kernel mode (guest believes it is in kernel mode)
     1. VMM examines every instruction guest is about to execute by reading a few instructions ahead of program counter
     2. Non-special-instructions run natively
     3. Special instructions translated into new set of instructions that perform equivalent task (for example changing the flags in the VCPU)

BUILDING BLOCKS - BINARY TRANSLATION



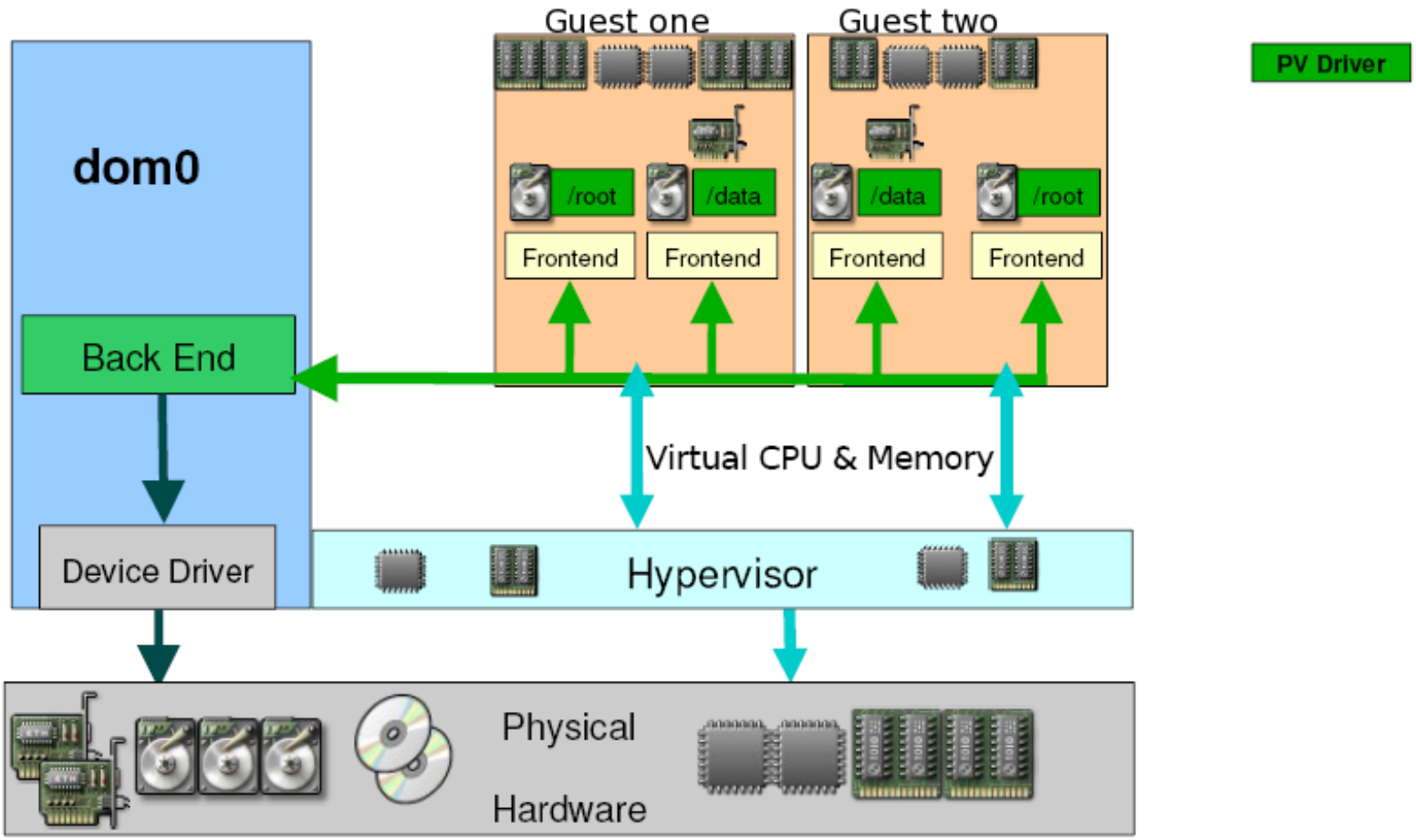
BUILDING BLOCKS - BINARY TRANSLATION

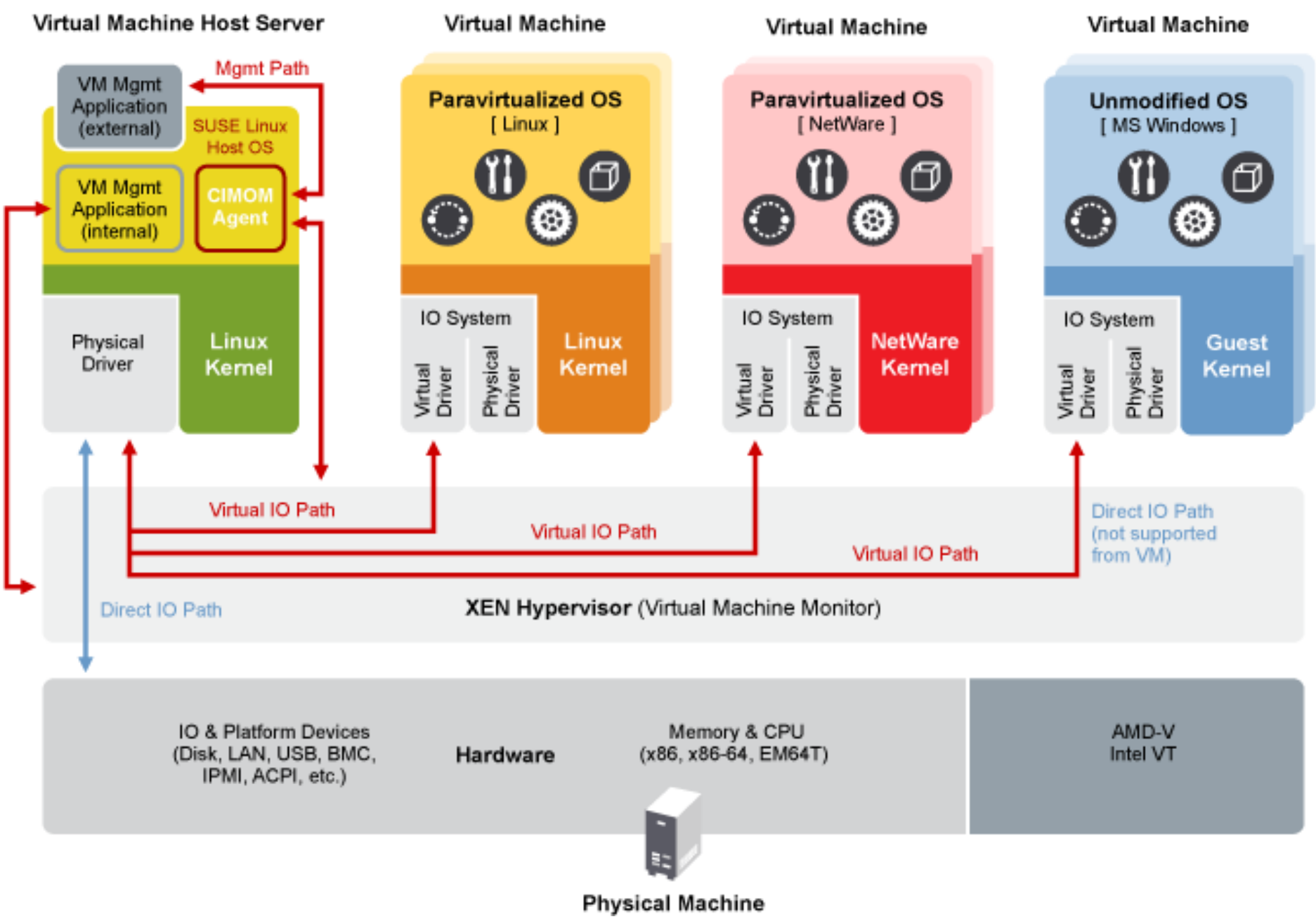
- Implemented by translation of code within VMM
- Code reads native instructions dynamically from guest, on demand, generates native binary code that executes in place of original code
- Performance of this method would be poor without optimizations
  - Products like VMware use caching
    - Translate once, and when guest executes code containing special instruction cached translation used instead of translating again
    - Testing showed booting Windows XP as guest caused 950,000 translations, at 3 microseconds each, or 3 second (5 %) slowdown over native

BUILDING BLOCKS - PARAVIRTUALIZATION

- Does not *really* fit the definition of virtualization – VMM not presenting an exact duplication of underlying hardware
  - But still useful!
  - VMM provides services that guest must be modified to use
  - Leads to increased performance
  - Less needed as hardware support for VMs grows
- Xen, leader in paravirtualized space, adds several techniques
  - For example, clean and simple device abstractions
    - Efficient I/O
    - Good communication between guest and VMM about device I/O
    - Each device has circular buffer shared by guest and VMM via shared memory

Xen Para-virtualization Architecture




Virtual Machine Host Server
Virtual Machine
Virtual Machine
Virtual Machine
Mgmt Path
VM Mgmt Application (external)
SUSE Linux Host OS
VM Mgmt Application (internal)
CIMOM Agent
Physical Driver
Linux Kernel
Paravirtualized OS [Linux]
IO System
Virtual Driver
Physical Driver
Linux Kernel
Paravirtualized OS [NetWare]
IO System
Virtual Driver
Physical Driver
NetWare Kernel
Unmodified OS [MS Windows]
IO System
Virtual Driver
Physical Driver
Guest Kernel
Virtual IO Path
Virtual IO Path
Virtual IO Path
Direct IO Path (not supported from VM)
Direct IO Path
XEN Hypervisor (Virtual Machine Monitor)
IO & Platform Devices (Disk, LAN, USB, BMC, IPMI, ACPI, etc.)
Hardware
Memory & CPU (x86, x86-64, EM64T)
AMD-V Intel VT
Physical Machine

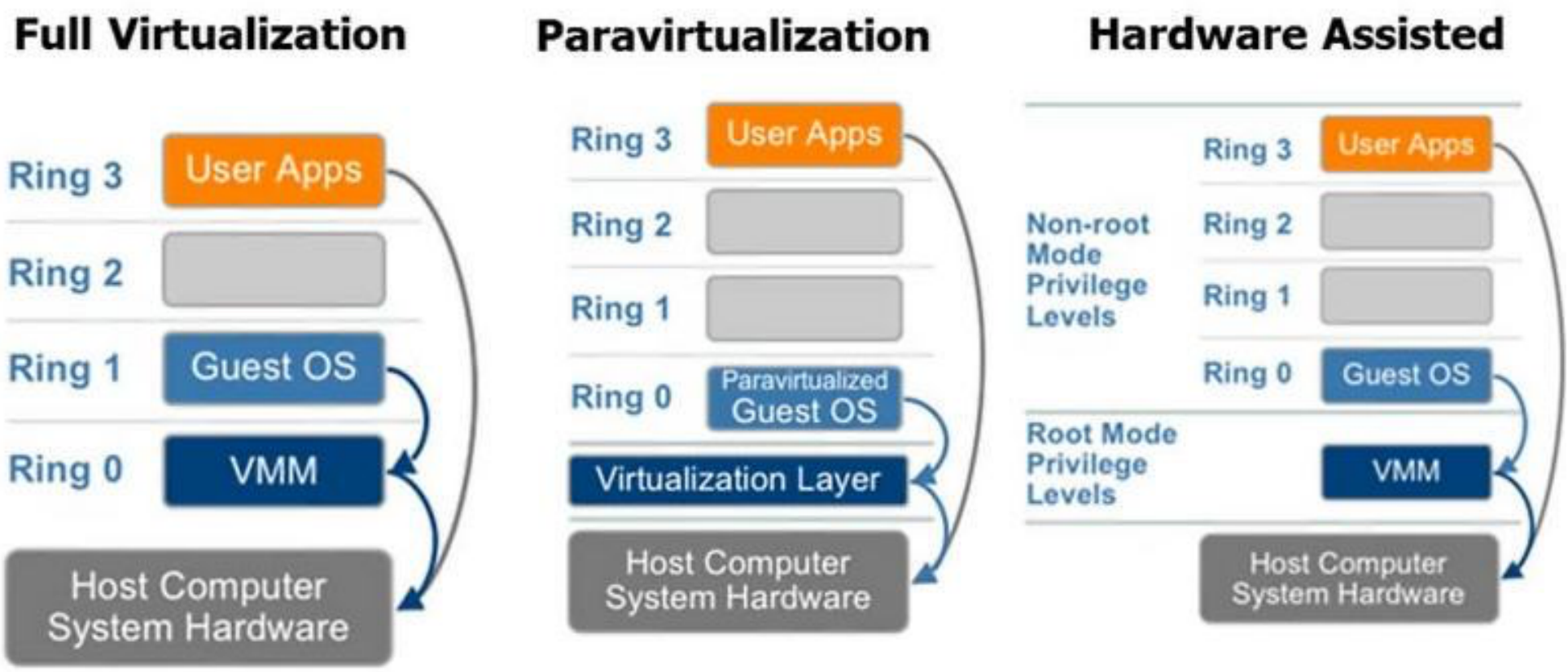
BUILDING BLOCKS - PARAVIRTUALIZATION

- Guest uses hypercall (call to hypervisor)
- Paravirtualization allowed virtualization of older x86 CPUs (and others) without binary translation
- Guest had to be modified to use run on paravirtualized VMM
- But on modern CPUs Xen no longer requires guest modification -> no longer paravirtualization

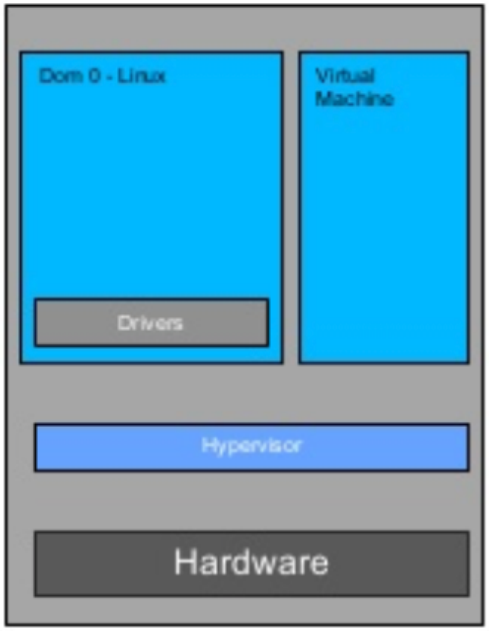
BUILDING BLOCKS - HARDWARE ASSISTANCE

- All virtualization needs some HW support
- More support -> more feature rich, stable, better performance of guests
- Intel added new **VT-x** instructions in 2005 and AMD the **AMD-V** instructions in 2006
  - CPUs with these instructions remove need for binary translation
  - Generally define more CPU modes – “guest” and “host”
  - VMM can enable host mode, define characteristics of each guest VM, switch to guest mode and guest(s) on CPU(s)
  - In guest mode, guest OS thinks it is running natively, sees devices (as defined by VMM for that guest)
    - Access to virtualized device, priv instructions cause trap to VMM
    - CPU maintains VCPU, context switches it as needed
- HW support for Nested Page Tables, DMA, interrupts as well over time

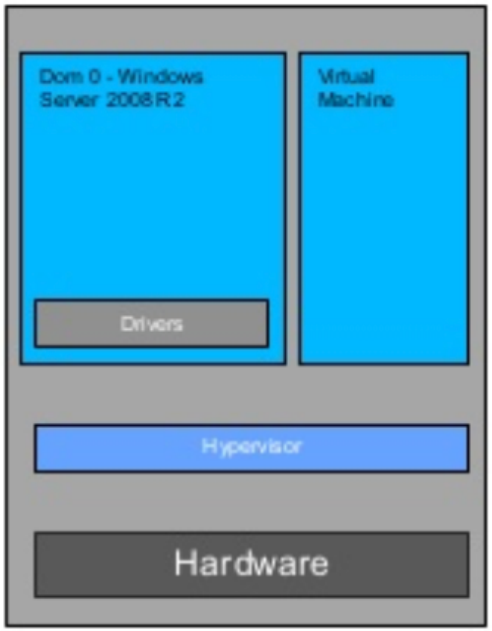
Architectural Comparison



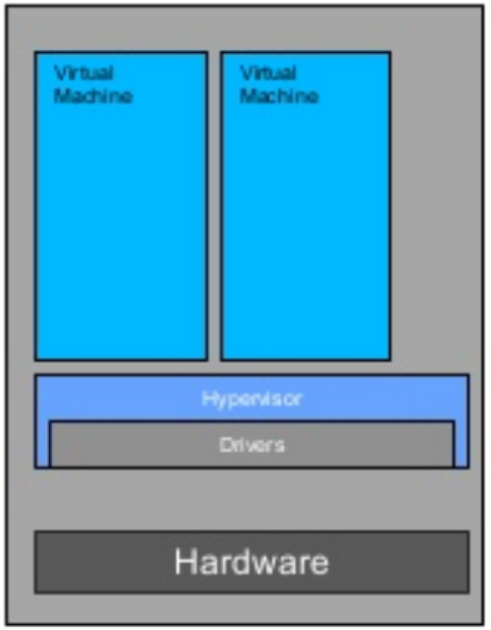
XEN VS HYPER-V VS ESXI ARCHITECTURE



XenServer



Hyper-V

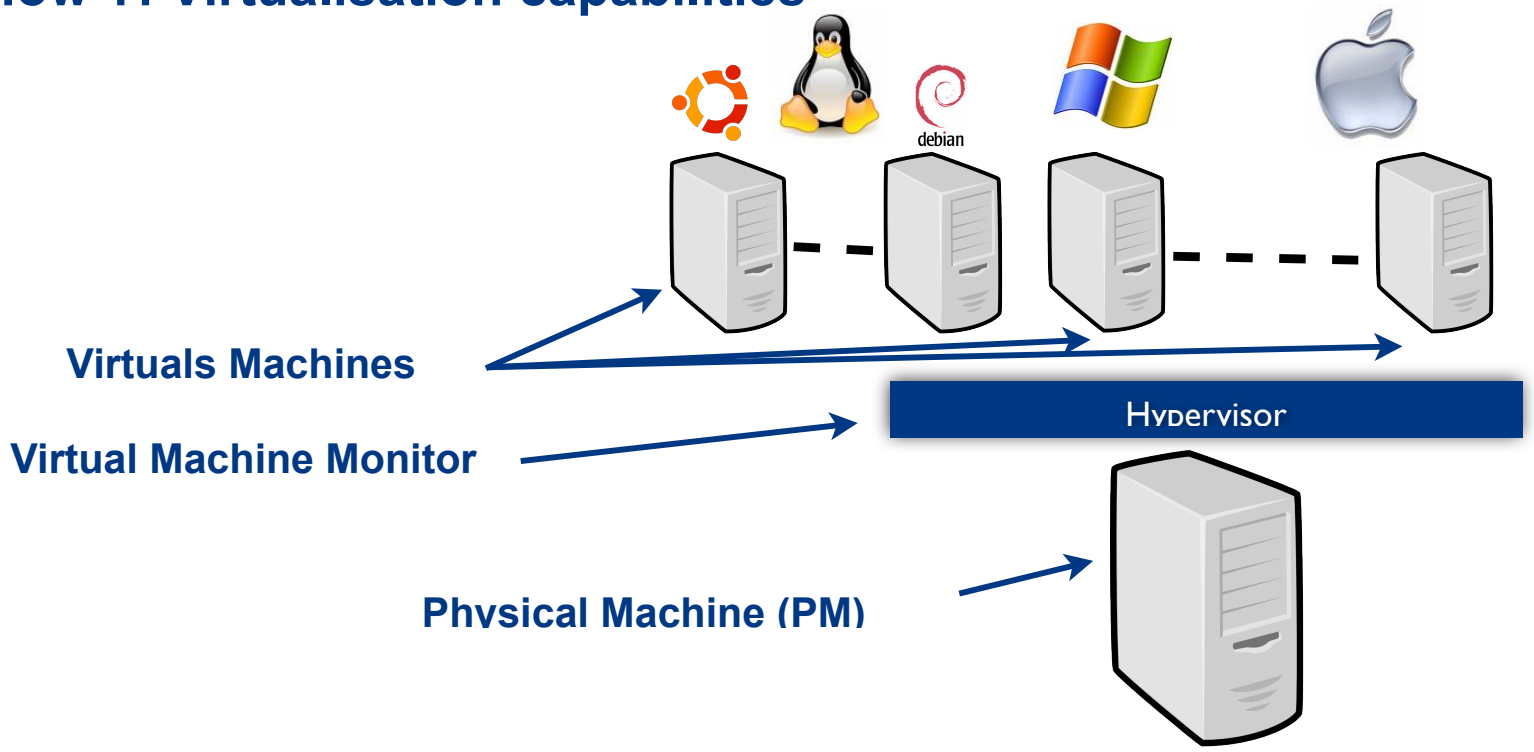


ESXi

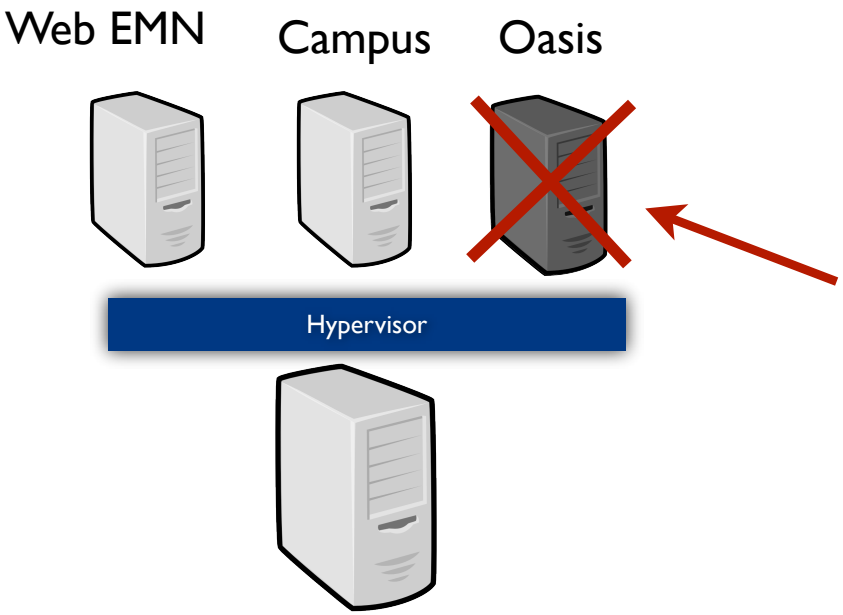
BENEFITS AND FEATURES

- **Consolidation (virtualization effect) :**
  - Consolidating to virtual machines reduces the number of running nodes

    So ernegy consumption
  - Reduces hardware costs while providing more efficient node
- **How ?: Virtualisation capabilities**



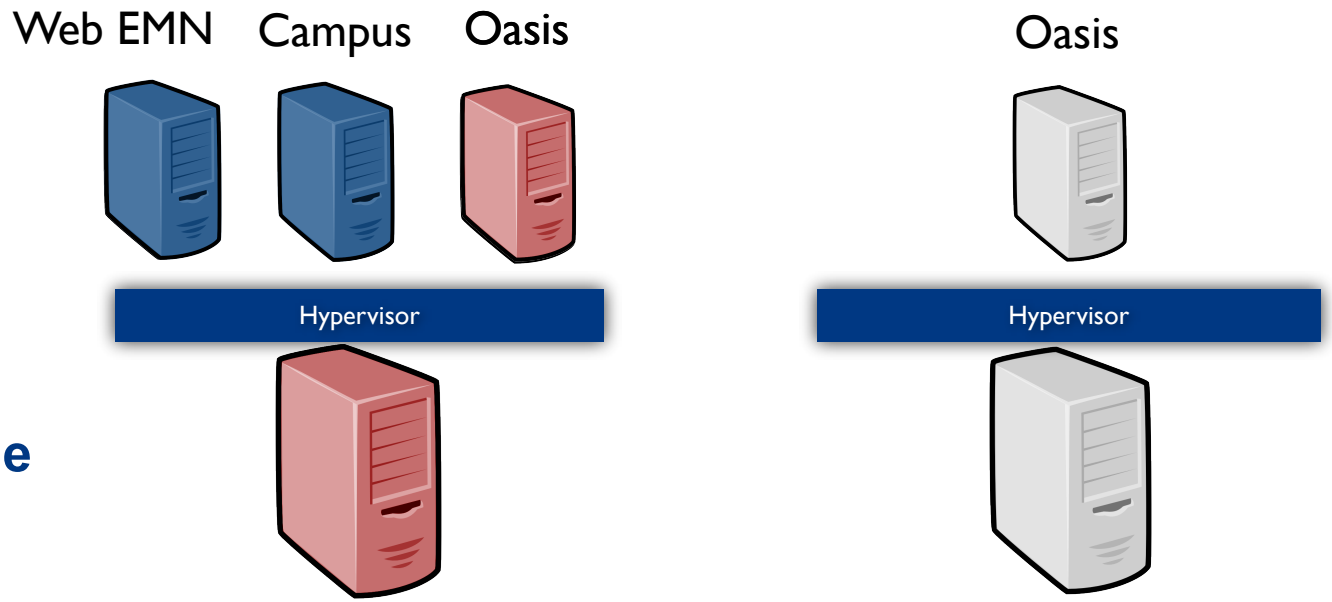
VIRTUALIZATION CAPABILITIES (1/2)



- **Isolation (security between VM)**
- **suspend/resume/reboot (maintenance)**

VIRTUALIZATION CAPABILITIES (2/2)

- **Live migration (load-balancing)**
- **High Availability(downtime ~ 60 ms)**



- **Dynamic Consolidation :**
  - The resources are allocated depending on the VM needs
  - VMs are mixed to be hosted on a reduced number of nodes
  - Servers unused can be turned off
  - VMs are remixed when it is necessary

BENEFITS AND FEATURES

- Host system protected from VMs, VMs protected from each other
  - I.e. A virus less likely to spread
  - Sharing is provided though via shared file system volume, network communication
- Freeze, **suspend**, running VM
  - Then can move or copy somewhere else and **resume**
  - Snapshot of a given state, able to restore back to that state
    - Some VMMs allow multiple snapshots per VM
  - **Clone** by creating copy and running both original and copy
- Great for OS research, better system development efficiency
- Run multiple, different OSes on a single machine
  - **Consolidation**, app dev, …

BENEFITS AND FEATURES

- **Templating** – create an OS + application VM, provide it to customers, use it to create multiple instances of that combination
- **Live migration** – move a running VM from one host to another!
  - No interruption of user access

- All those features taken together -> **cloud computing**
  - Using APIs, programs tell cloud infrastructure (servers, networking, storage) to create new guests, VMs, virtual desktops

CHAPTER 4.2
INSIDE VIRTUALIZATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

INSIDE VIRTUALIZATION

- Now look at operating system aspects of virtualization
  - CPU scheduling, memory management, I/O, storage, and unique VM migration feature
    - How do VMMs schedule CPU use when guests believe they have dedicated CPUs?
    - How can memory management work when many guests require large amounts of memory?

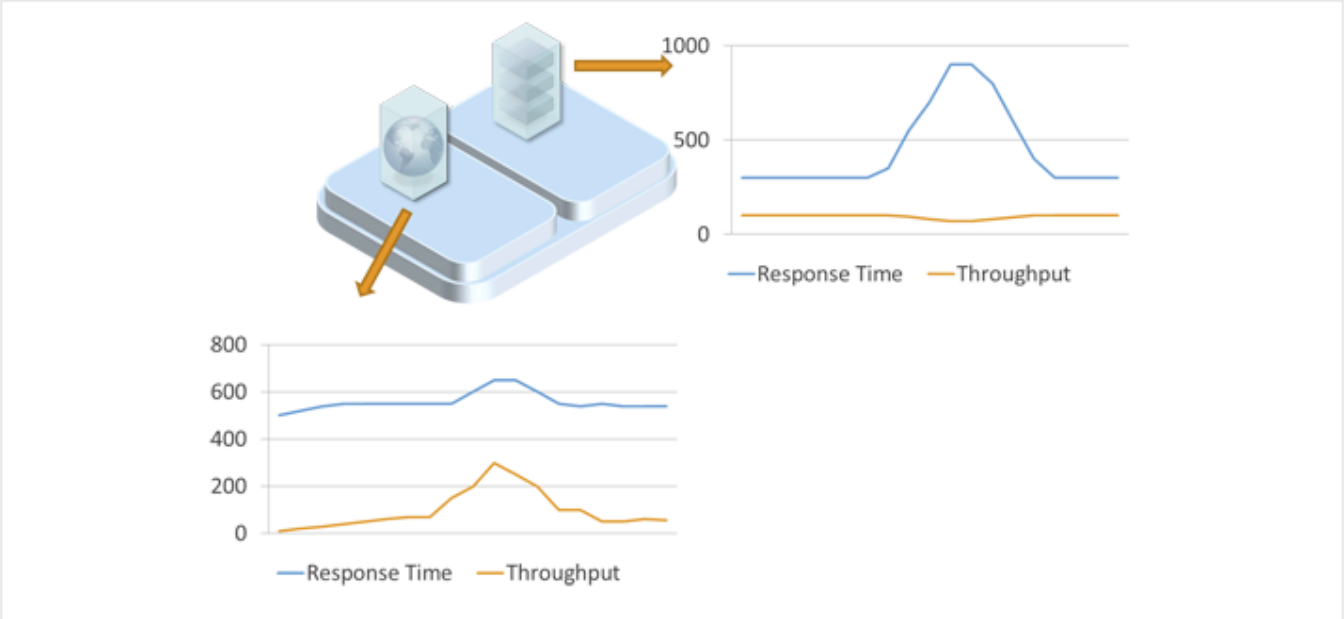
CPU

- Even single-CPU systems act like multiprocessor ones when virtualized
  - One or more virtual CPUs per guest
- Generally VMM has one or more physical CPUs and number of threads to run on them
  - Guests configured with certain number of VCPUs
    - Can be adjusted throughout life of VM
  - When enough CPUs for all guests -> VMM can allocate dedicated CPUs, each guest much like native operating system managing its CPUs
  - Usually not enough CPUs -> CPU **overcommitment**
    - VMM can use standard scheduling algorithms to put threads on CPUs
    - Some add fairness aspect

CPU

- Cycle stealing by VMM and oversubscription of CPUs means guests don't get CPU cycles they expect
  - Consider timesharing scheduler in a guest trying to schedule 100ms time slices -> each may take 100ms, 1 second, or longer
    - Poor response times for users of guest
    - Time-of-day clocks incorrect
  - Some VMMs provide application to run in each guest to fix time-of-day and provide other integration features

Why running isolated VMs also hides the root cause of performance problems

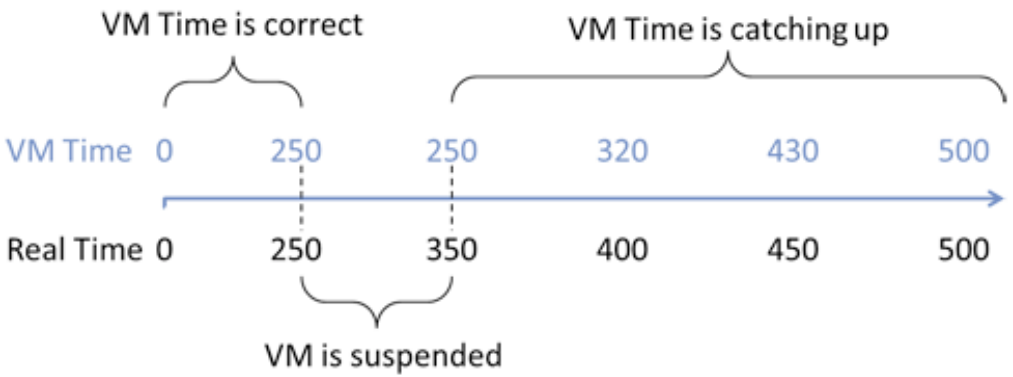


A spike on one application can impact the performance of another even though they run in separate VMs

A problem caused by an application running in one VM, only shows up in the second VM

We need to correlate the performance measurements from both applications using monitoring data at the hypervisor layer

When the hypervisor suspends a guest system, it is just like an OS suspending one process to execute another. But where the OS is responsible for time keeping, the guest system won't usually know when it is suspended and can't account for this when tracking time



The time as seen by the VM and real time can deviate due to VM suspension and the way systems track time

Windows and Linux systems track time by counting the periodically triggered interrupts.

A suspended guest can't count interrupts

The hypervisor records interrupts and replays them when the VM is resumed.

After resuming, a lot of queued interrupts are processed in quick succession.

As the interrupts represent time slices, time effectively speeds up!

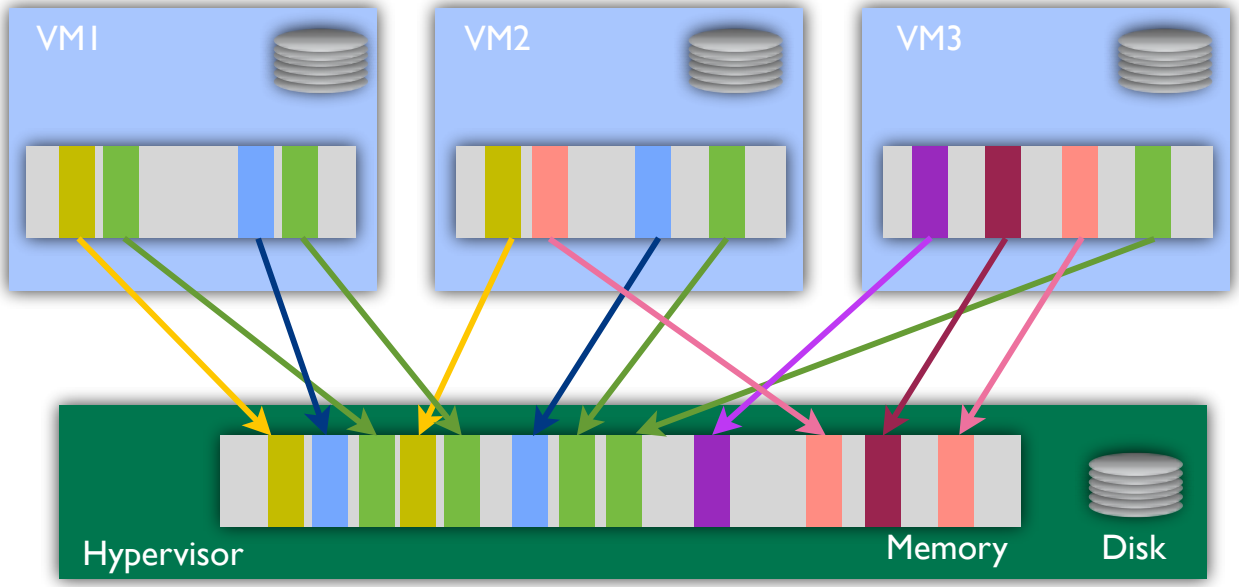
CPU : CO-SCHEDULING

- Schedule all VCPU in the same time
  - In 4 CPU processor, a 4 VCPU VM needs that all CPU are free.
  - In 4 CPU processor, it’s hard to find a scheduling with
    - 1 VM (2VCPU) 1 VM (3VCPU) 1 VM (4VCPU)
- However …
  - Necessary for inside VM for SPINLOCK

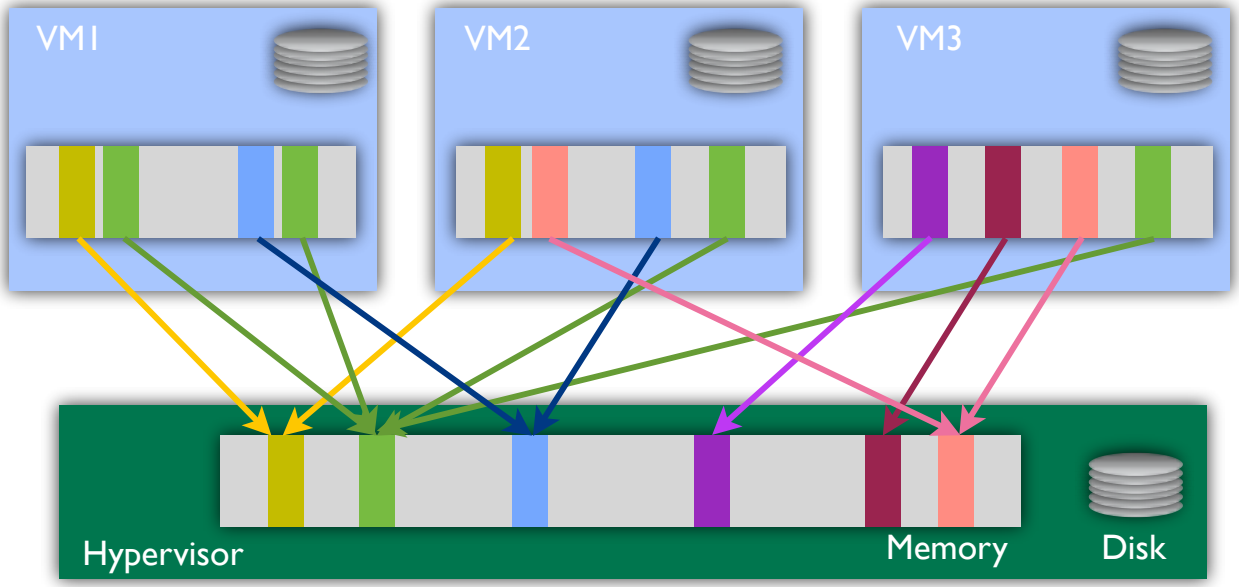
RESSOURCE HOT PLUG : VCPU

- Ability to hot plug VCPU !
  - To dynamically change the number of VCPU inside VM (for exemple for workload change)
- The main Idea :
  - Administrator need a VM with 2VCPU
  - Start a VM with 8 VCPU, and declare that 6 VCPU are breaks.
  - When need, change the state of a VCPU from break to ready.
- The OS inside VM must be able to accepte this technic.
- Working also for RAM

Memory overcommitment



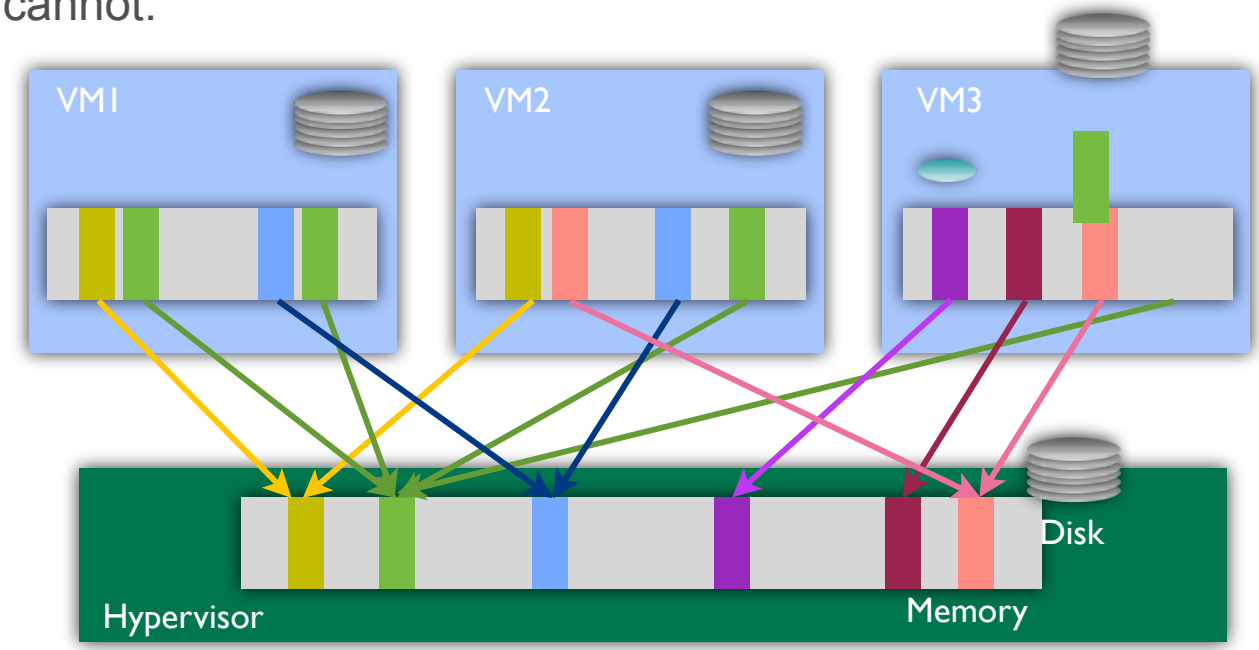
Content-Based sharing



- The concept of transparent page sharing was first proposed in the Disco system [1997-17]

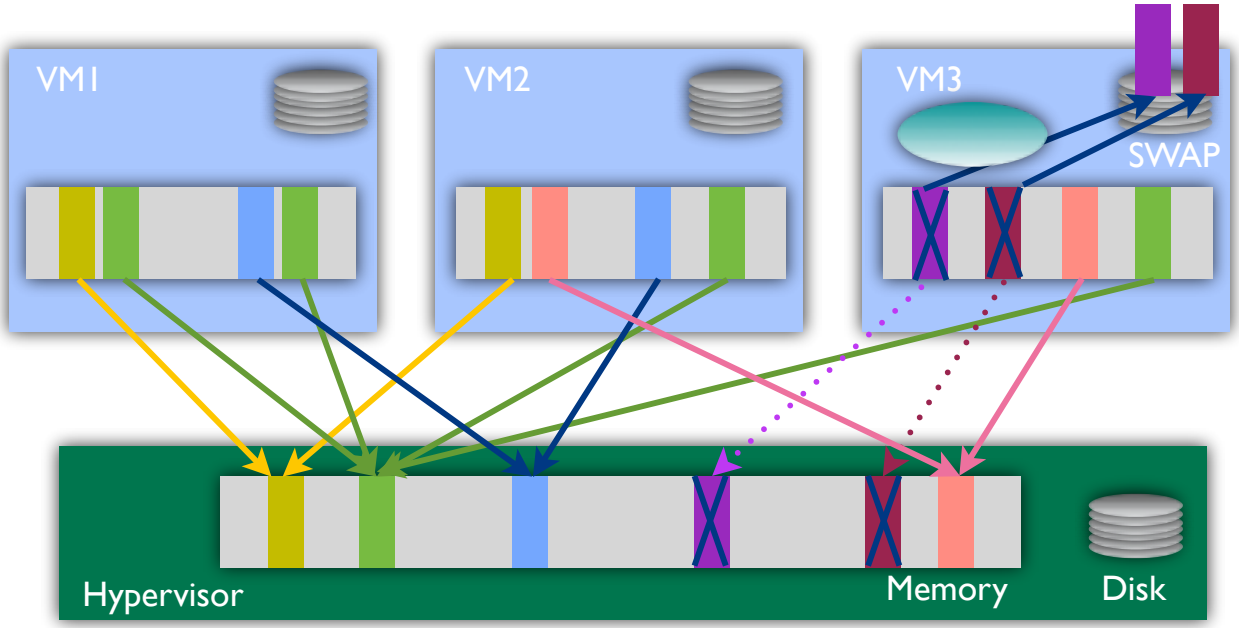
Ballooning

The guest can make an informed decision about what to swap, whereas the hypervisor cannot.



Ballooning

The guest can make an informed decision about what to swap, whereas the hypervisor cannot.

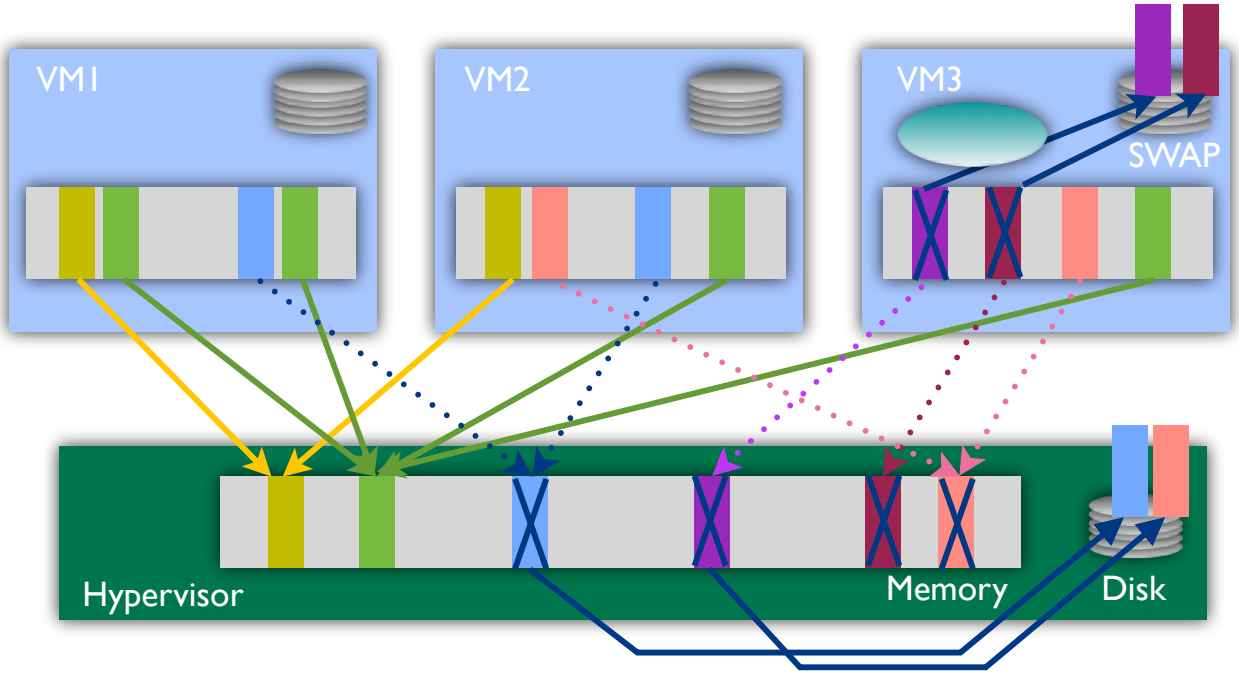


Inflate the balloon by allocating guest memory and pinning (can not swapped to disk) the underlying memory pages.

The hypervisor can reclaim those pinned pages and reassign it to another VM.

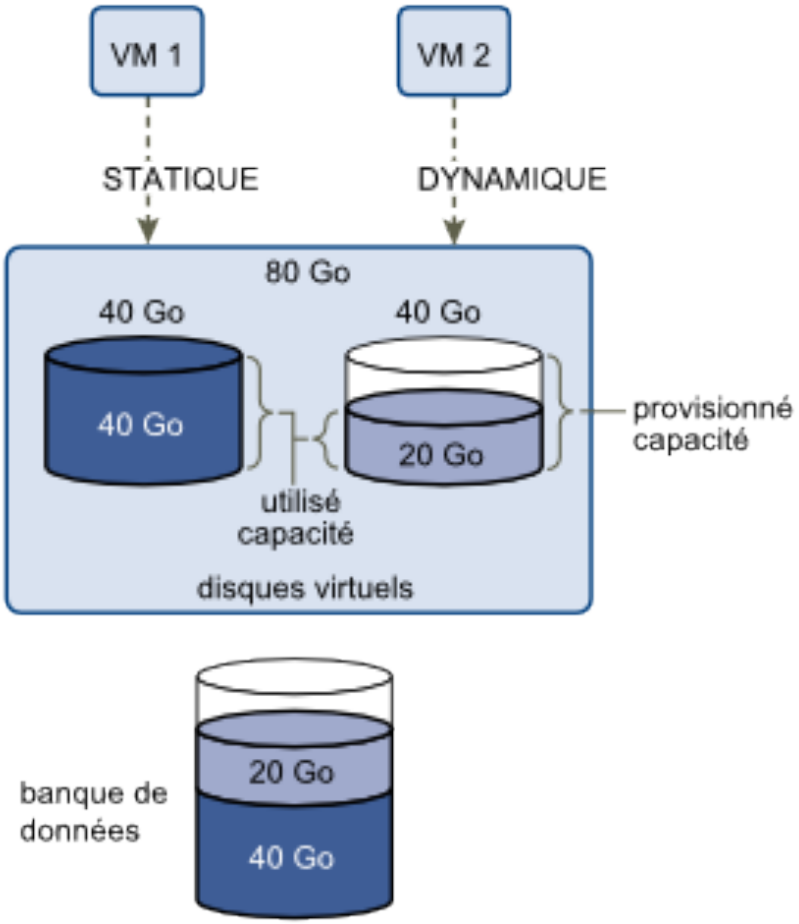
A guest with a balloon might be forced to swap other pages.

Hypervisor Swapping



MEMORY

- **Effective only if it is complemented by algorithms that ensure that the VMs resident on each physical server contain a significant amount of sharable pages.**
- **Memory Buddies [2010] Goals :**
  - Analyze the memory contents of multiple VMs to determine **sharing potential** then find **more compact VM placement**
  - Evaluation show that “sharing aware” placement has the ***potential*** to significantly improve memory usage (20 VM on 4 servers).
  - ***Invasive*** system (nucleus component into each virtual machine)
- **Sharing-Aware Algorithms for Virtual Machine Colocation [2012]**
  - ***simulation*** with (124 VM on 25 servers) and ***offline***
- ***CBS Challenge :***
  - Transparent Page Sharing with Large Pages, Effects of Memory Randomization, Sanitization and Page Cache on Memory Deduplication ...
- **Dynamic consolidation with resource sharing aware**


VM 1
VM 2
STATIQUE
DYNAMIQUE
80 Go
40 Go
40 Go
40 Go
20 Go
provisionné
capacité
utilisé
capacité
disques virtuels
banque de
données
20 Go
40 Go

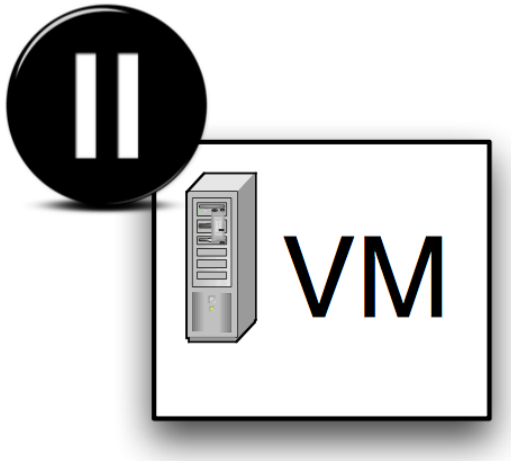
MIGRATION

- Relocate VM from one physical host to another
  Complete encapsulation → no OS support needed
- Transfer VM state over the network
  - Processor state (CPU registers)
  - Hardware devices state (hardware registers)
  - Memory content
    (Possibly disk content)

PURE STOP-AND-COPY

- Simplest approach
- Suspend source VM on source host
- Copy all VM state over the network
- Resume source VM on destination host
- Used by the Internet Suspend/Resume project

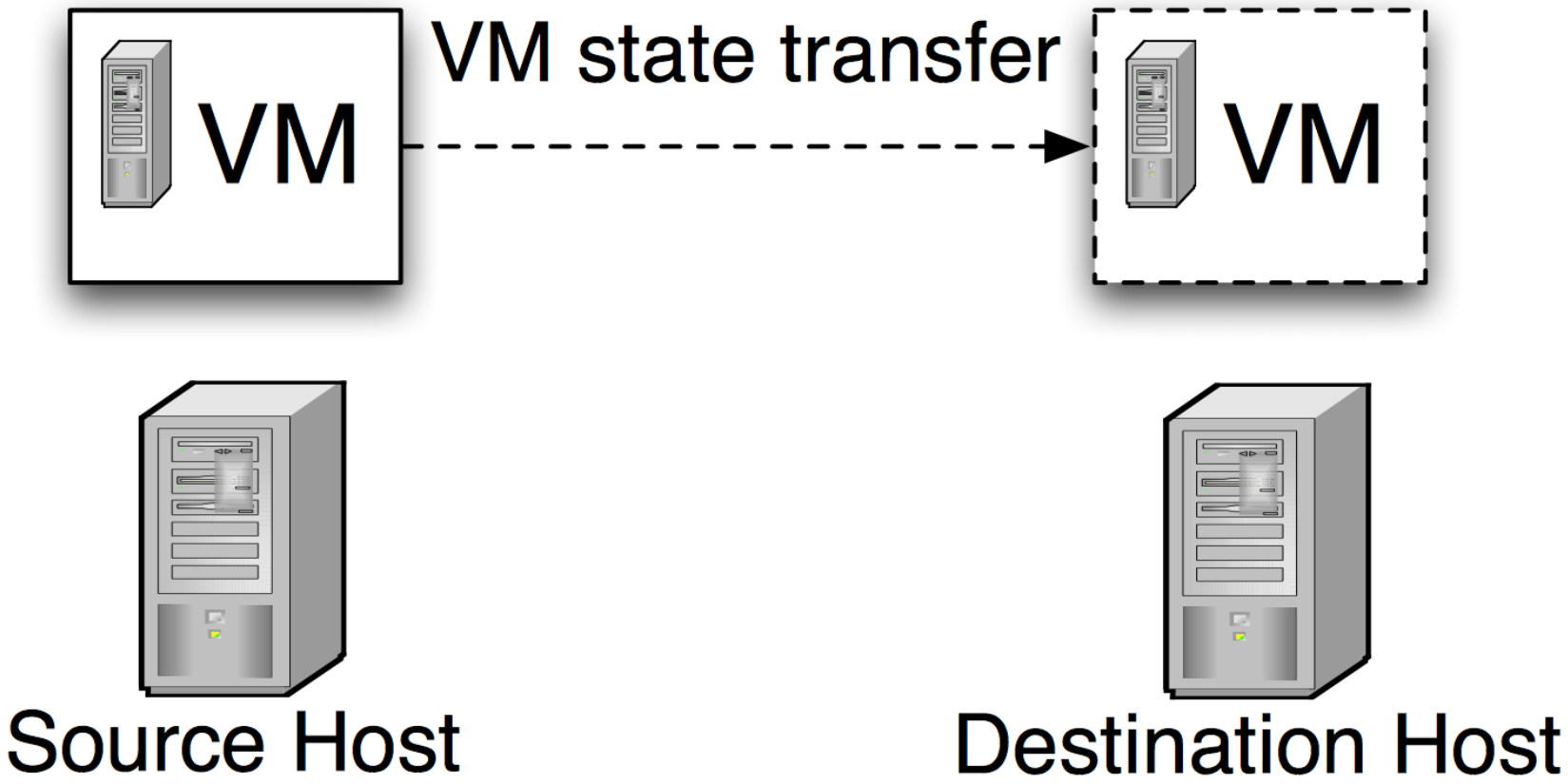
PURE STOP-AND-COPY



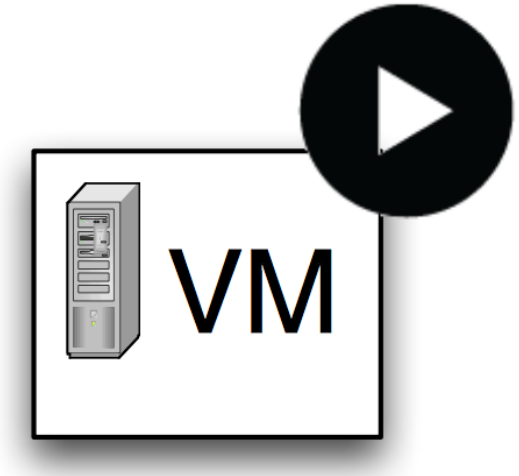
Source Host

Destination Host

PURE STOP-AND-COPY



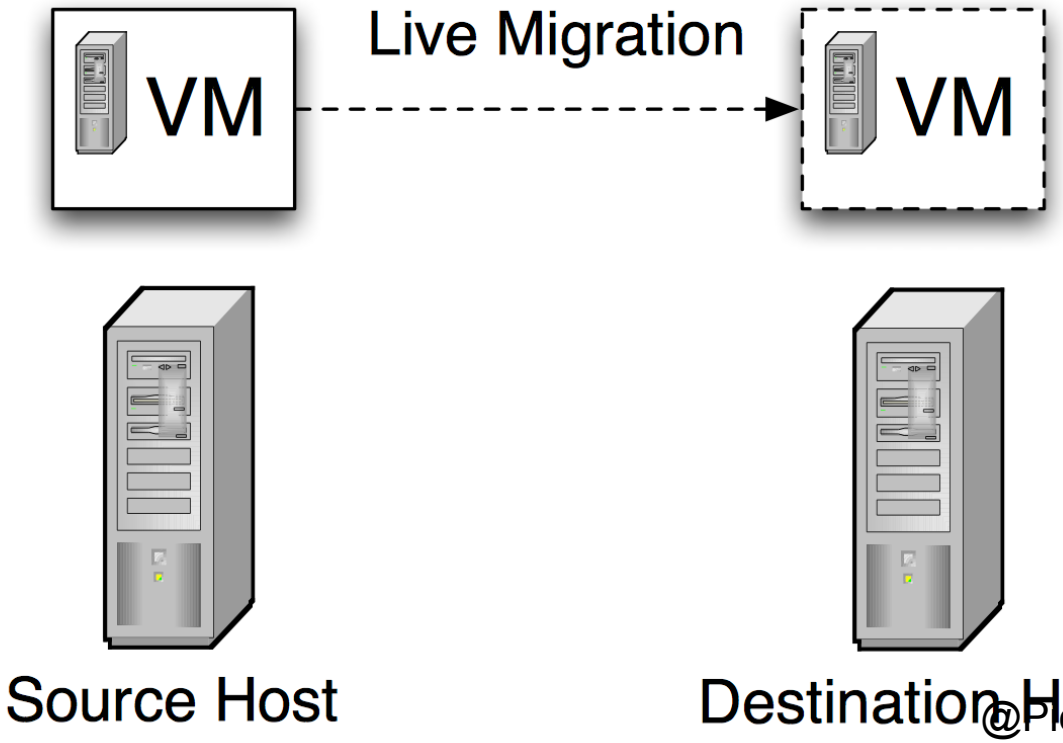
PURE STOP-AND-COPY



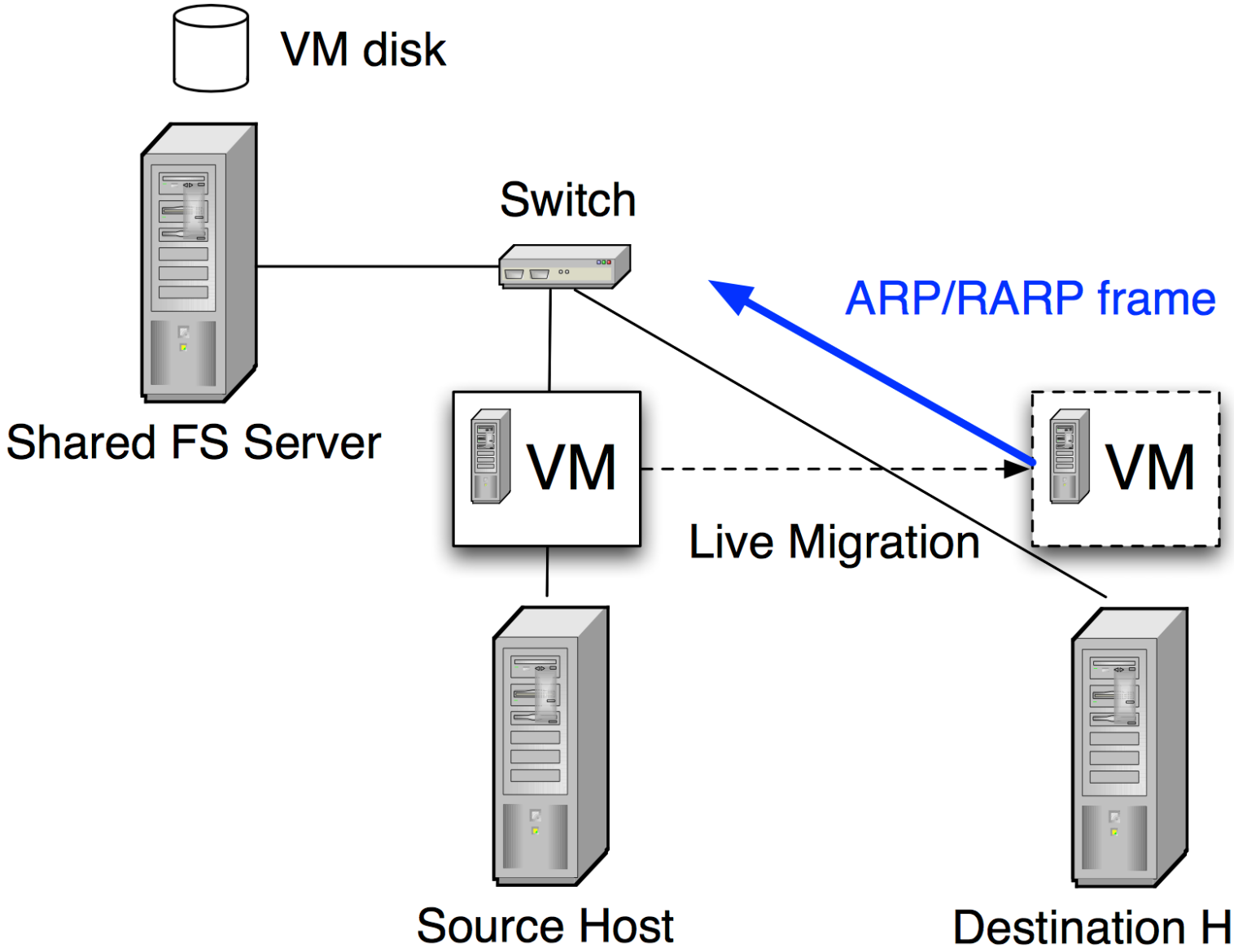
Source Host

Destination Host

- Problem with pure stop-and-copy:
  - long downtime
- Live migration
  - Minimize downtime (milliseconds)
  - Works by transferring state during execution



- Live migration initially proposed for LANs
- Clark et al., NSDI ’05 & Nelson et al., USENIX ’05
- Transfer from source host to destination host of the same LAN
- What about storage and network resources?
  - Shared storage (e.g. NFS) → no migration needed
  - Network traffic redirected with gratuitous ARP/RARP frames


VM disk
Switch
ARP/RARP frame
Shared FS Server
VM
VM
Live Migration
Source Host
Destination Host

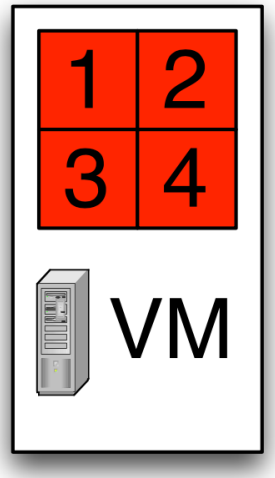
LIVE MIGRATION OF VIRTUAL MACHINES USE CASES

- Offers many advantages
  - Load balancing / Reduced energy consumption
  - Migrate VMs in case of hotspots
  - Consolidate VMs on a subset of nodes
  - Turn off unused physical nodes
  - Entropy (Jean-Marc Menaud)
- Transparent infrastructure maintenance
- Pro-active fault tolerance
  - Detect future faults from hardware events
  - Preemptively migrate on another node
  - Nagarajan et al., SC 07

PRE-COPY LIVE MIGRATION

- Traditional method used for migration of processes
- Iterative process
  - Copy all memory content to the destination host (while the VM continues running)
  - Do multiples iterations to copy modified memory pages during the previous period
  - When enough iterations have been done, stop the VM and
    - Copy the remaining modified memory pages
    - Copy the CPU and device state
- Resume VM on destination host
- **Method implemented by all production hypervisors**

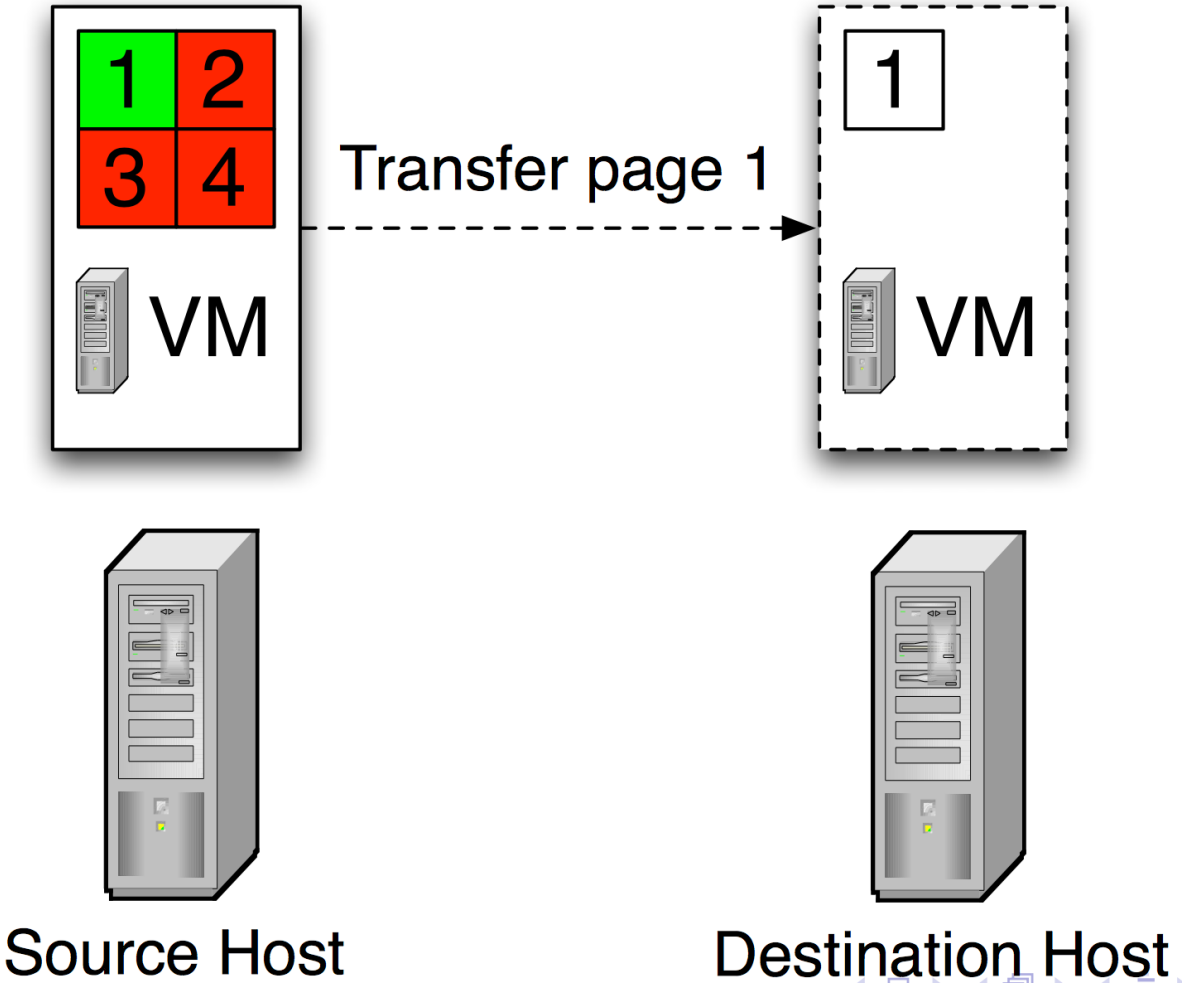
PRE-COPY LIVE MIGRATION



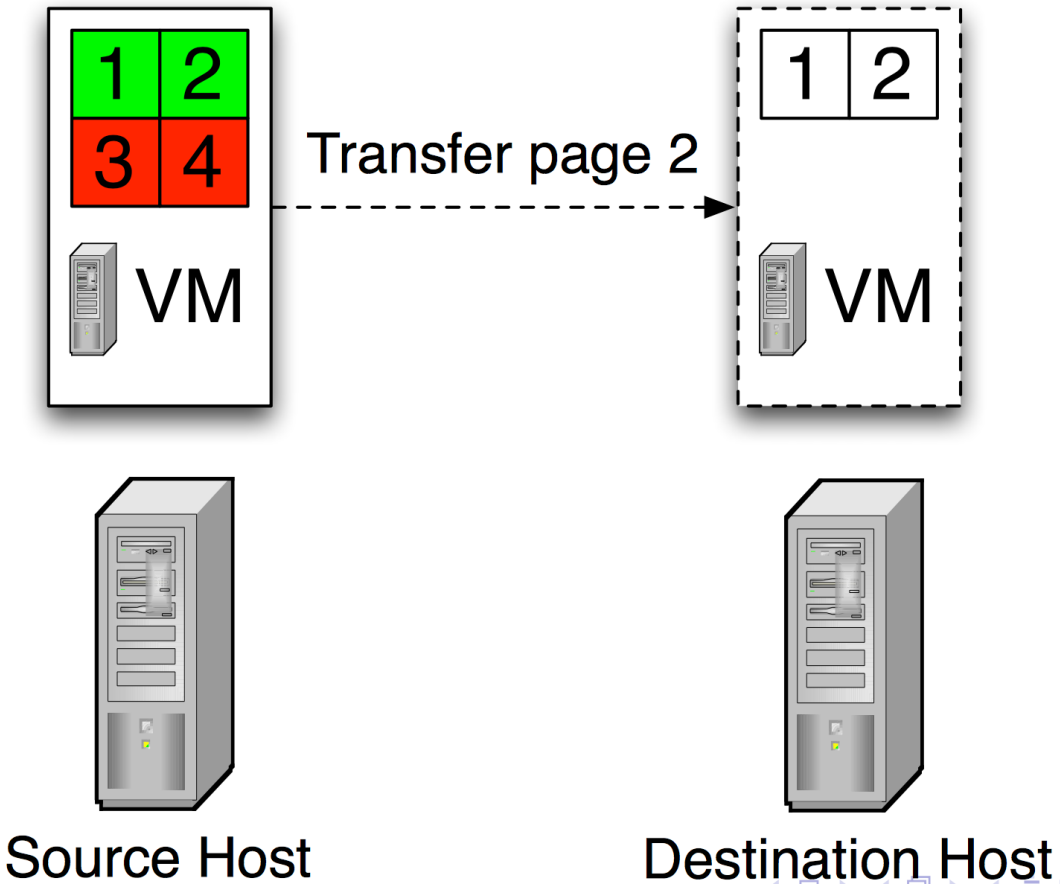
Source Host

Destination Host

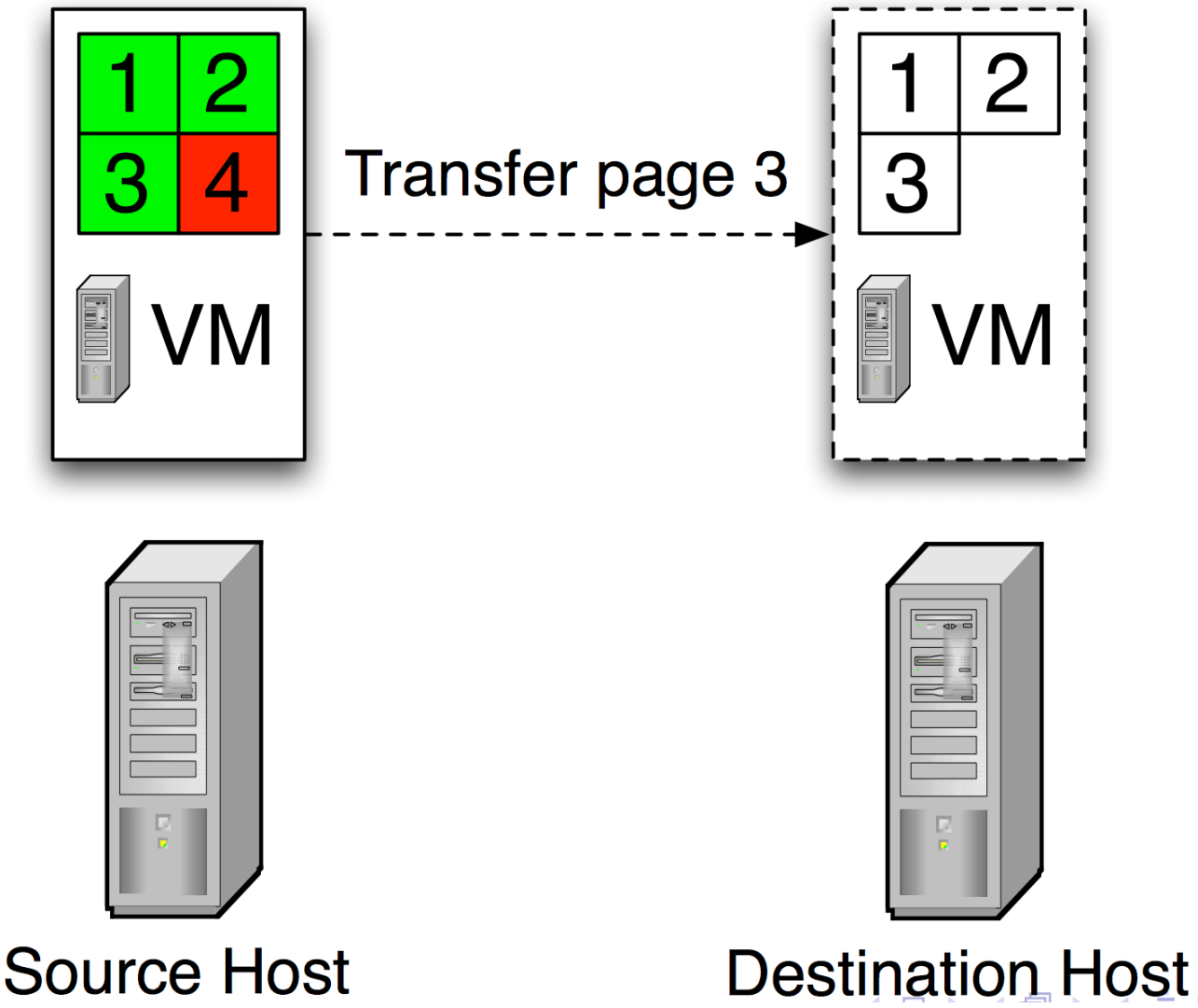
PRE-COPY LIVE MIGRATION



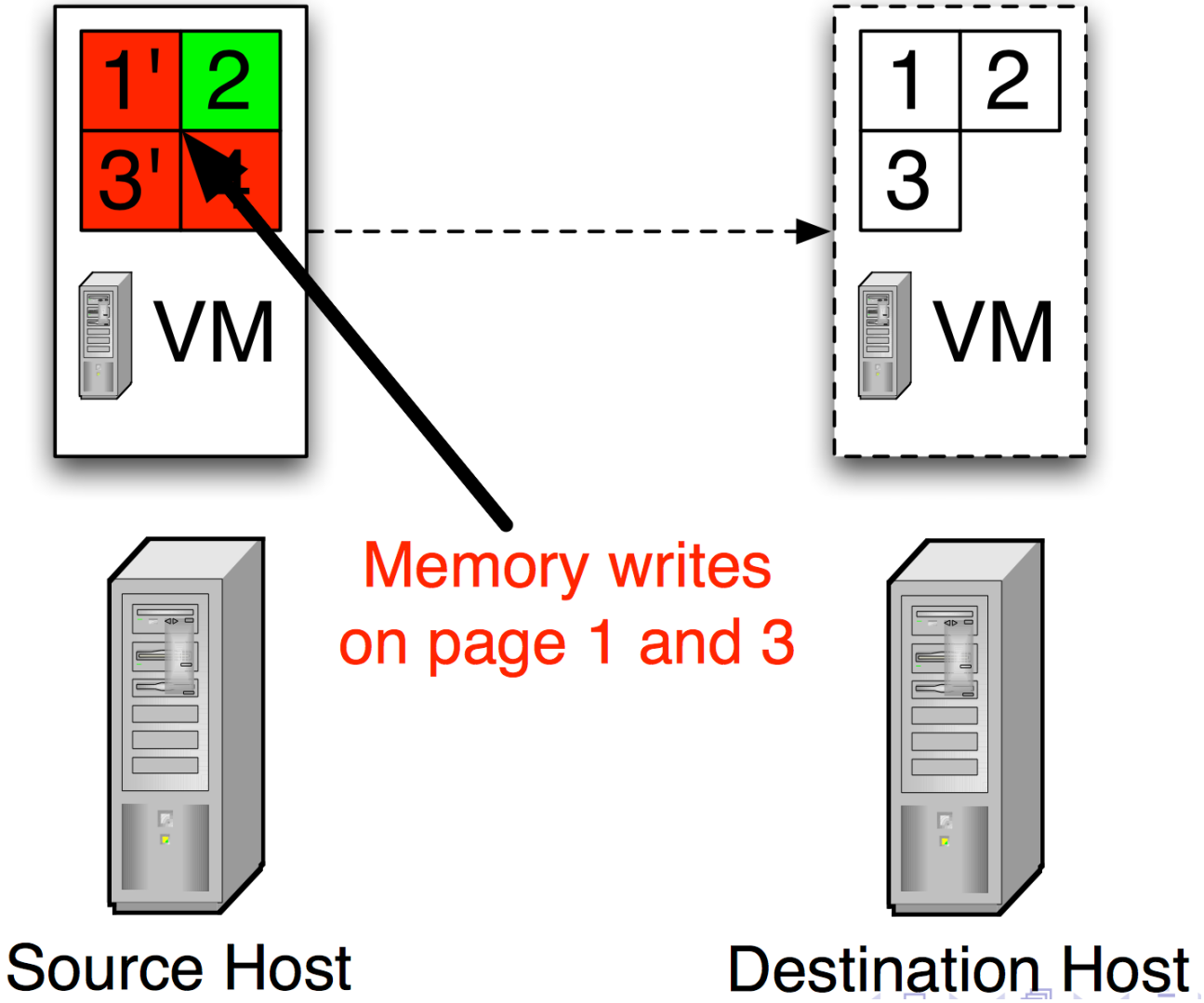
PRE-COPY LIVE MIGRATION



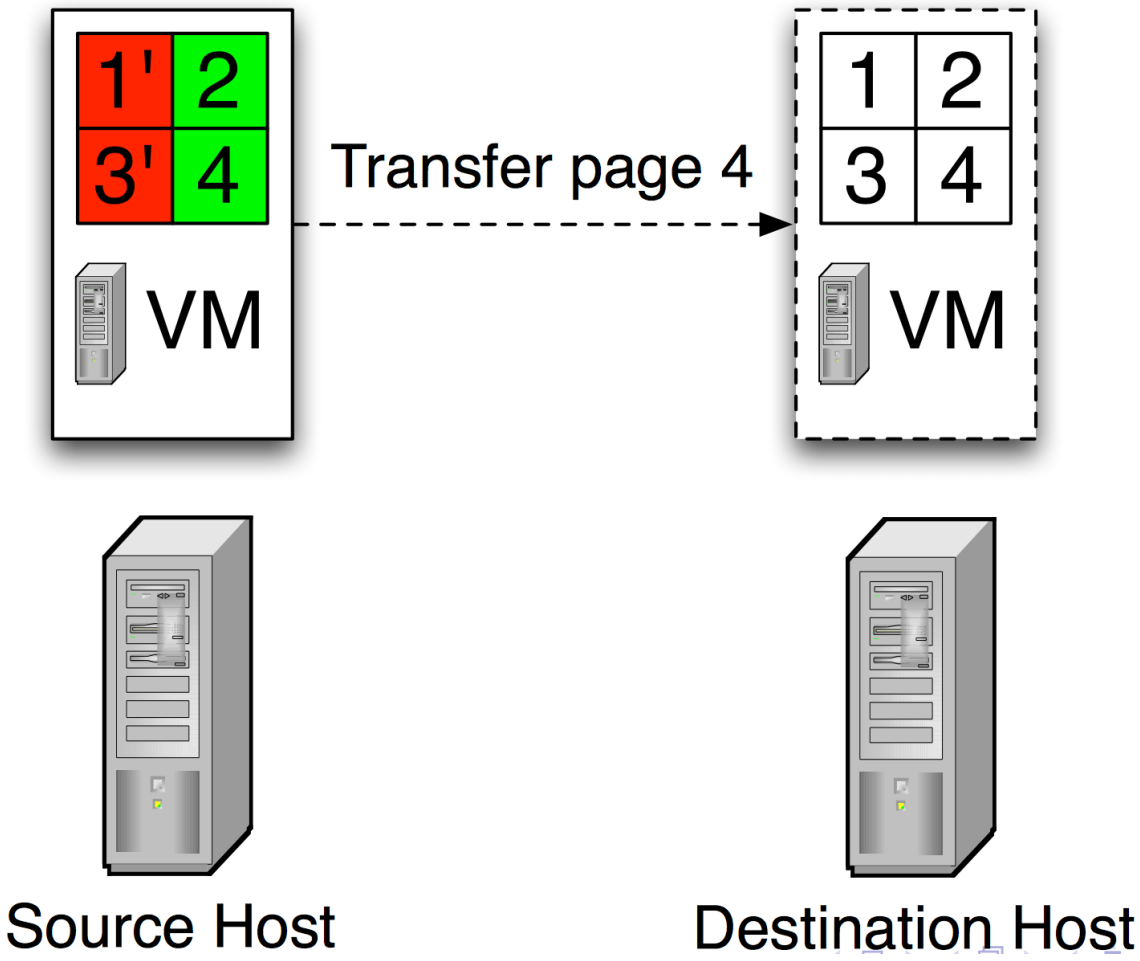
PRE-COPY LIVE MIGRATION



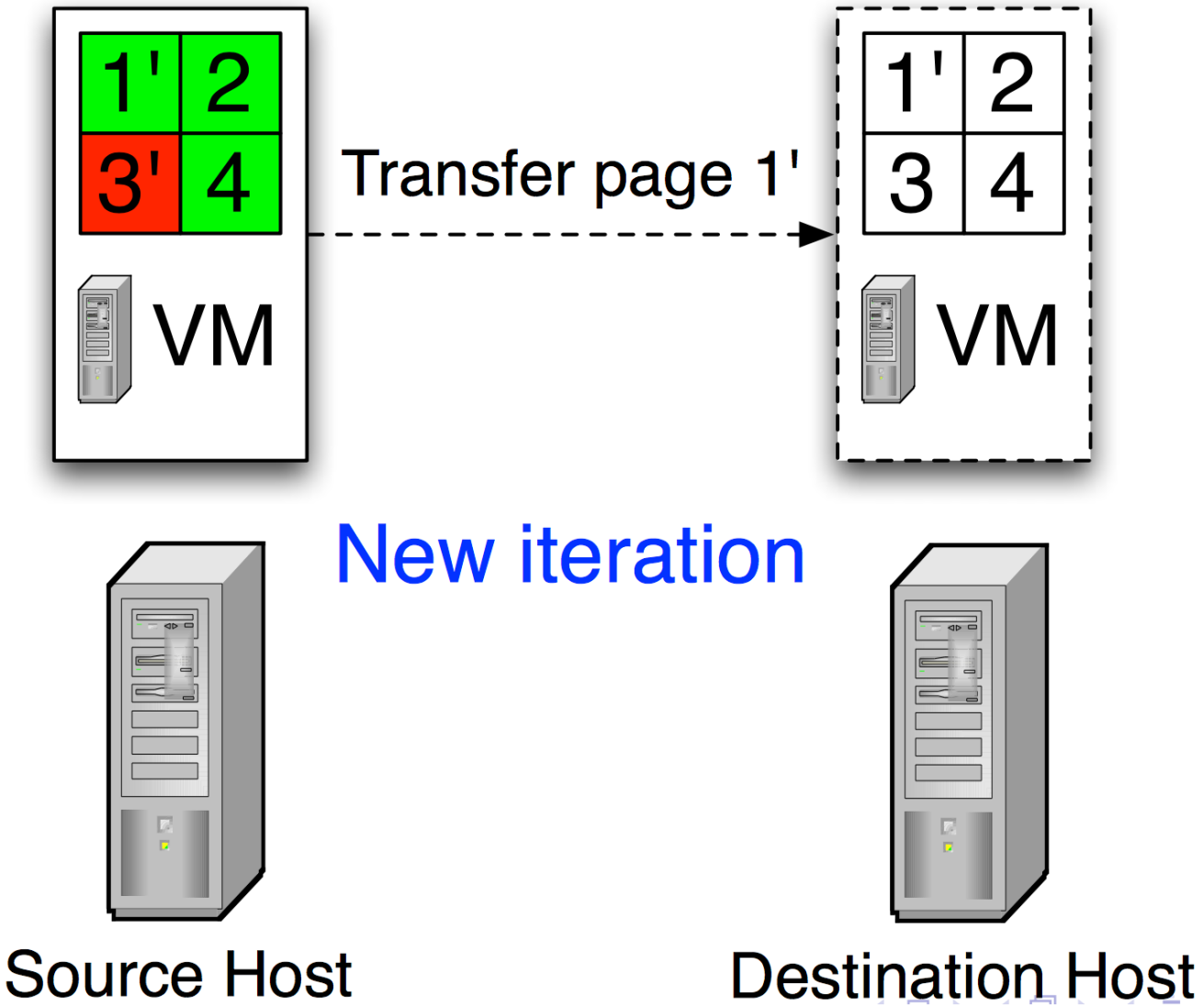
PRE-COPY LIVE MIGRATION



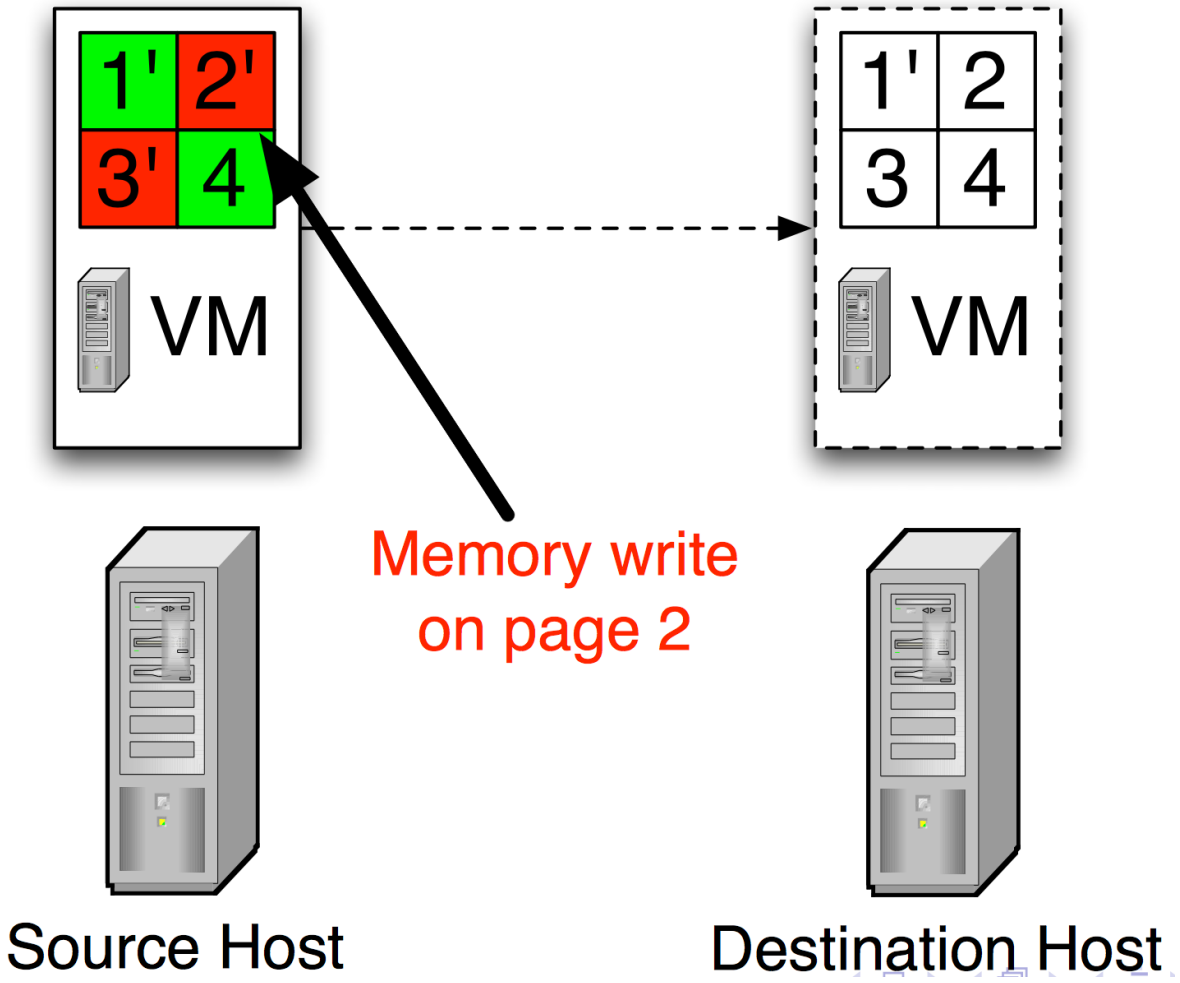
PRE-COPY LIVE MIGRATION



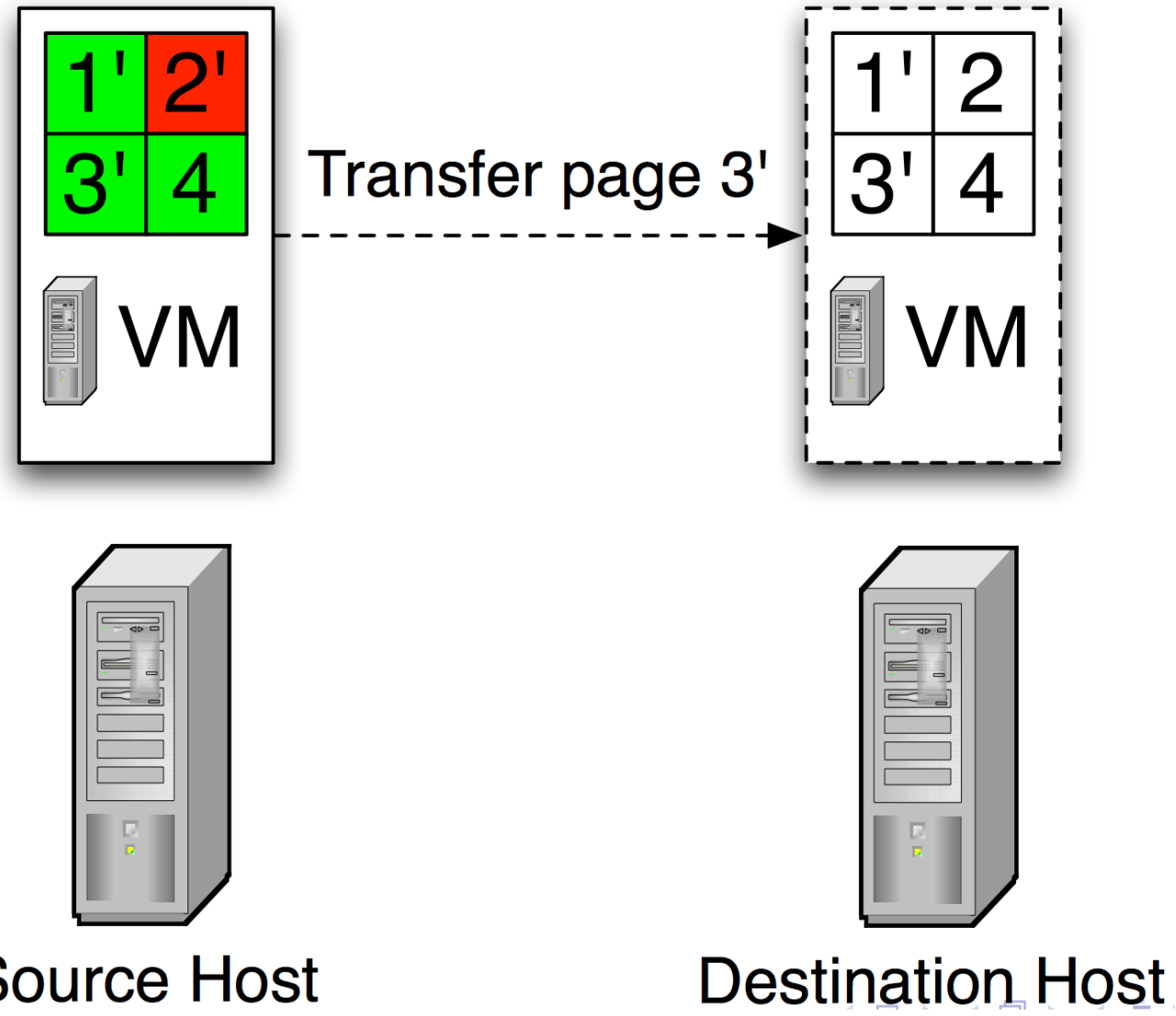
PRE-COPY LIVE MIGRATION



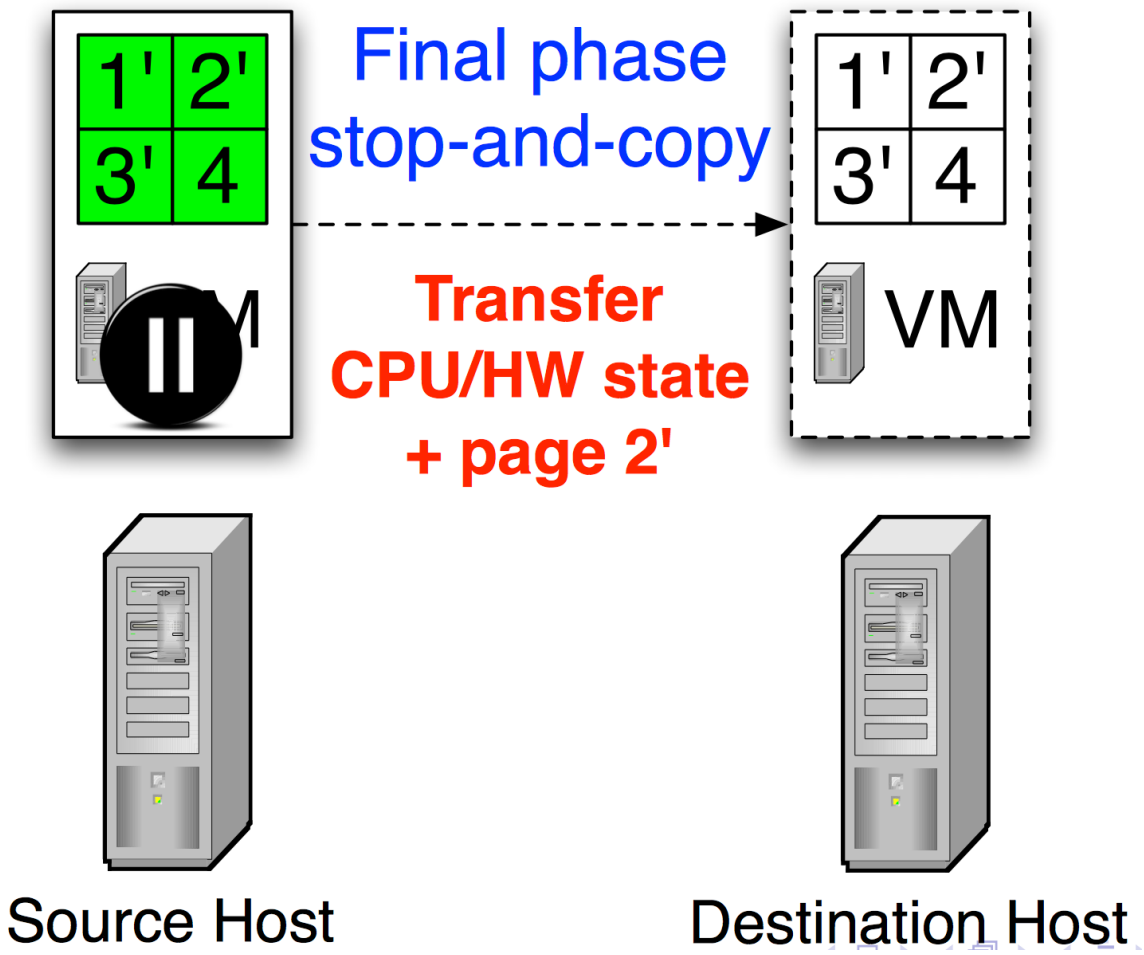
PRE-COPY LIVE MIGRATION



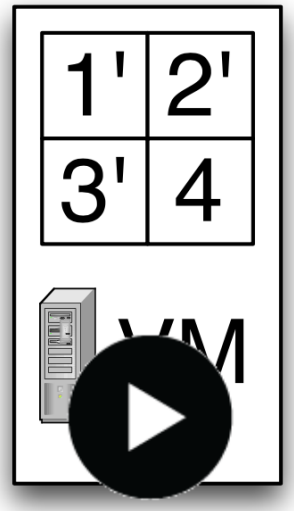
PRE-COPY LIVE MIGRATION



PRE-COPY LIVE MIGRATION



PRE-COPY LIVE MIGRATION



Source Host

Destination Host

STOP-AND-COPY PHASE

- Different behavior between Xen and KVM
- Xen: threshold values
  - Remaining pages under a threshold
  - OR Too many iterations
  - OR Too much data transferred
- KVM: estimated downtime
  - Administrator can specify maximum downtime
  - Default: 30 milliseconds
  - KVM estimates available bandwidth
  - Stops only when estimated downtime < maximum downtime
- Xen forces convergences of migration
- KVM trusts the admistrator or VM management software

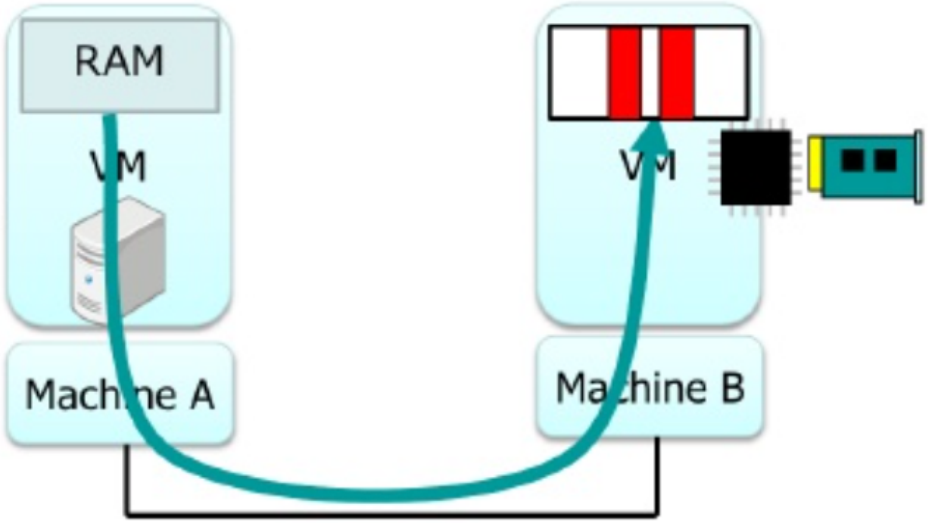
POST-COPY LIVE MIGRATION

- Pre-copy can present long downtime in the last phase
  - if the application modifies a large working set
  - if the available bandwidth is low
- Post-copy algorithm
  - Start by copying CPU and device state
  - Resume VM execution on the destination host
  - Fetch memory on demand when accessed
- Reduces downtime over pre-copy
- Can lower performance because of memory access latency
- KVM implementation: Takahiro Hirofuchi & Isaku Yamahata

POST-COPY LIVE MIGRATION

Postcopy live migration

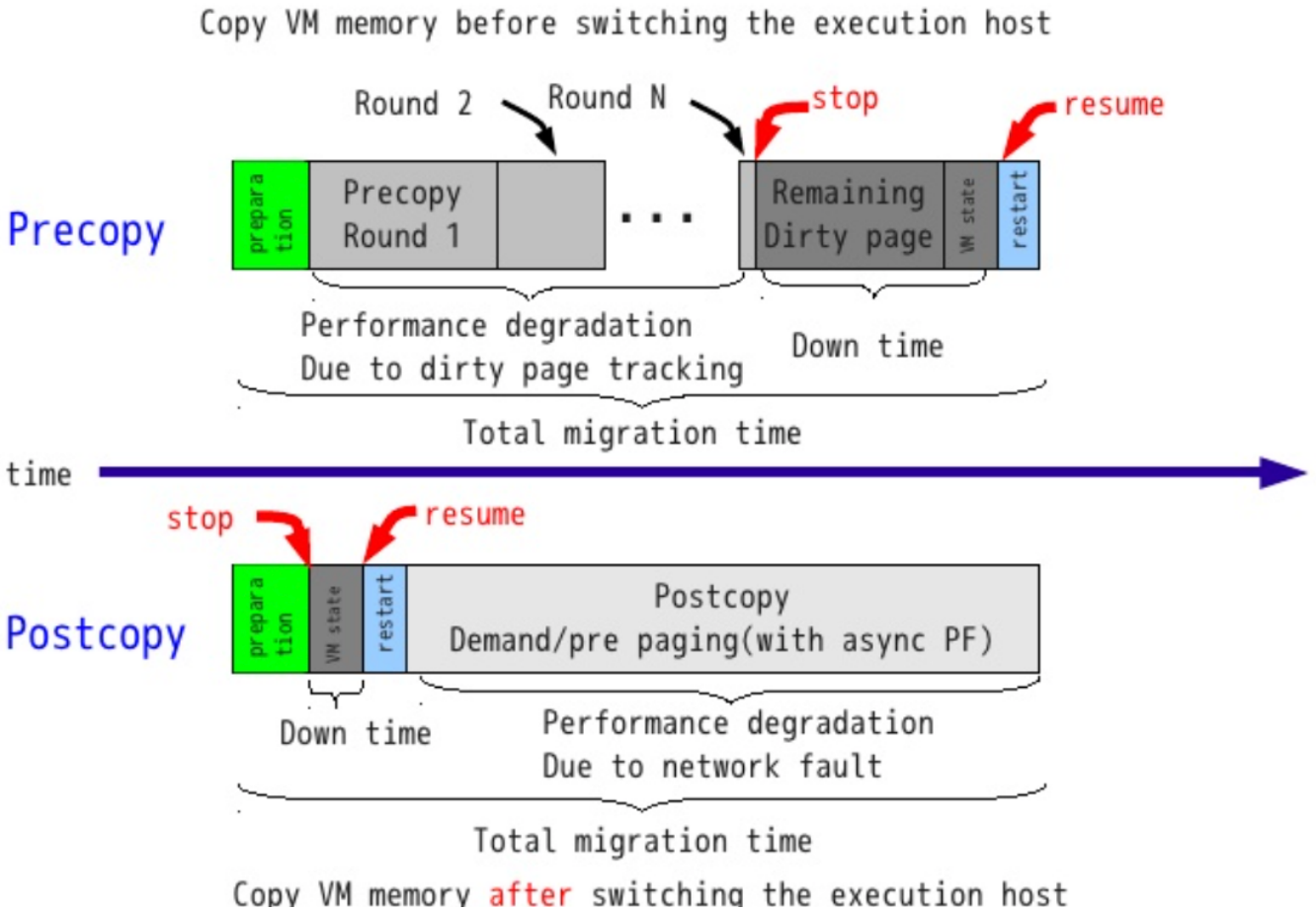
1. Stop VM
2. Copy non-memory VM state to destination
3. Resume VM at destination
4. Copy memory pages on-demand/backgroundly
   - Async PF can be utilized



Copy memory pages
- On-demand(network fault)
- background(precache)

POST-COPY LIVE MIGRATION

Total migration/down time



LIVE MIGRATION OPTIMIZATIONS

- Metrics to minimize
  - Total data transferred
  - Downtime
  - Total migration time
- Several approaches
  - Data Compression
  - Page Delta Transfer
  - Data Deduplication

DATA COMPRESSION

- Compress memory pages sent over the network
- Compress zero'd memory pages → available in KVM
  - Interesting for migration of Windows
- Use a compression algorithm (gzip, bzip2, lzo)
  - → KVM supports piping VM state to any executable
- Adaptive memory compression [Jin:2009]

PAGE DELTA TRANSFER

- Memory pages are 4 KB on x86
  - Modify 1 byte in the page → transfer 4 KB
- Delta transfer mechanism:
  - Keep copy of original page
  - Computer differences between original and new page
  - Send diff instead of full content

DATA DEDUPLICATION

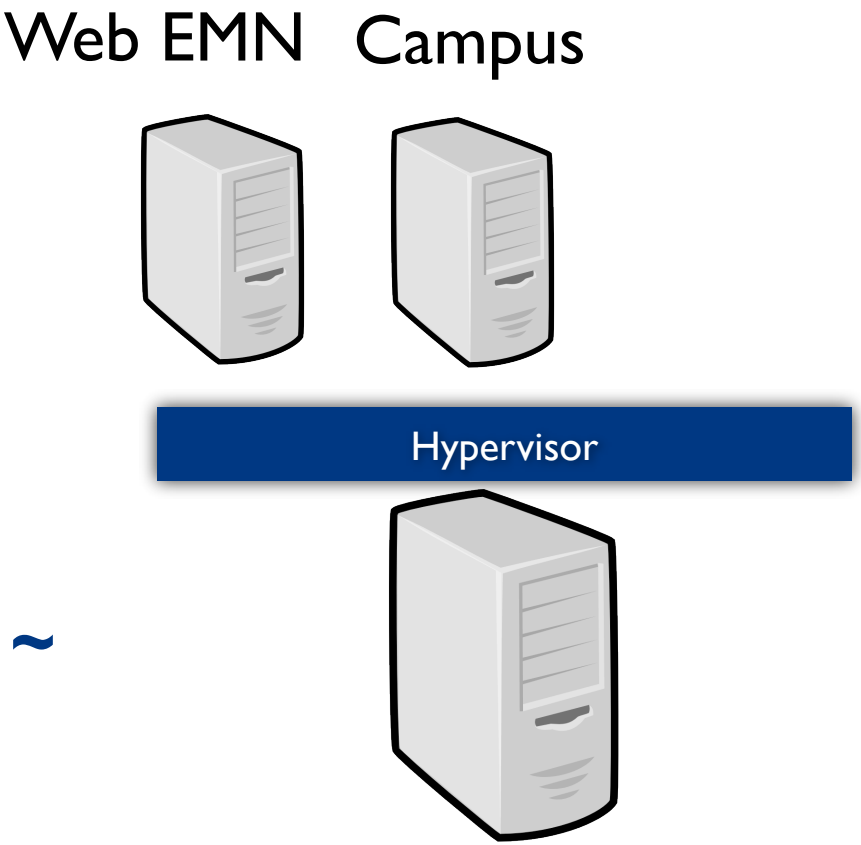
- VMs can contain identical data in multiple memory pages
- Remove duplicated memory pages
- Fast hash algorithm + full data comparison when match
  - Single-VM [Wood et al., VEE 2011]
  - Multi-VM on same host [Deshpande et al., HPDC 2011]
- Distributed approach for Multi-VM Multi-host
  - [Riteau et al., Euro-Par 2011]

CHAPTER 4.3
CONSOLIDATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

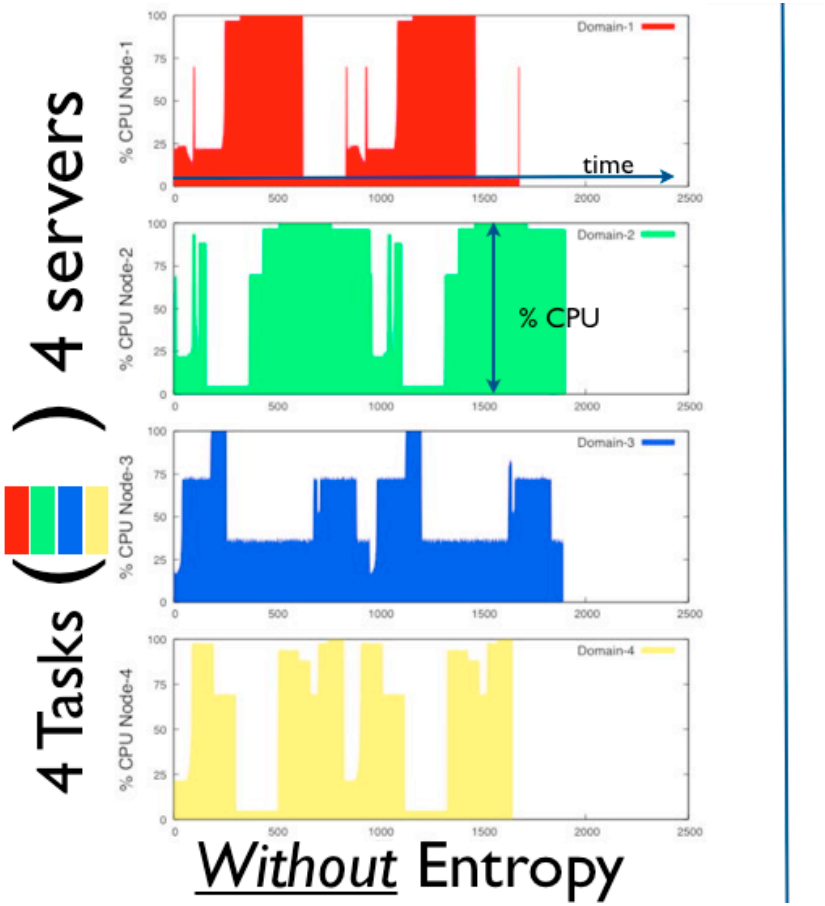
LIVE MIGRATION

- **Live migration (load-balancing)**
- **High Availability(downtime ~ 60 ms)**

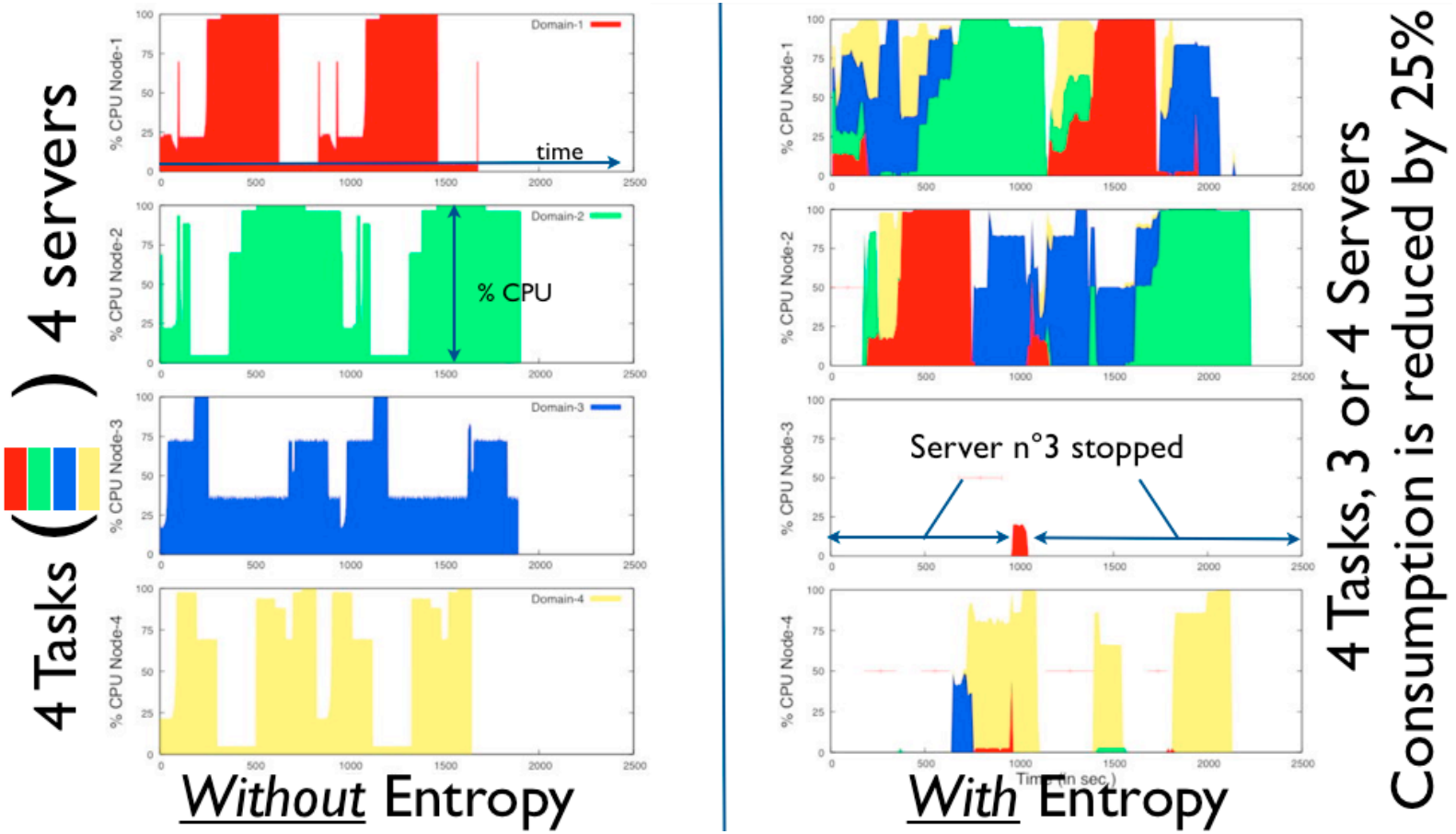


- **Dynamic Consolidation :**
  - The resources are allocated depending on the VM needs
  - VMs are mixed to be hosted on a reduced number of nodes
  - Servers unused can be turned off
  - VMs are remixed when it is necessary

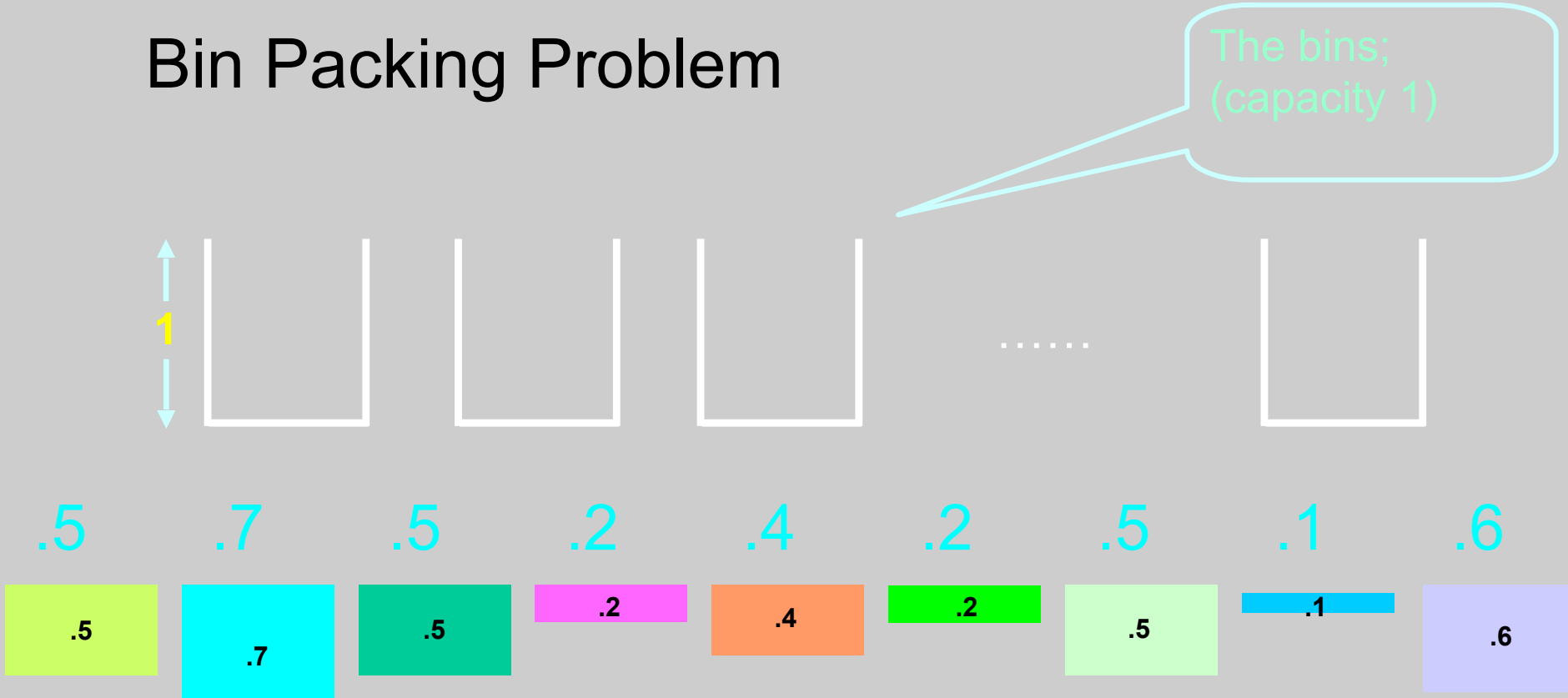
ENERGY USE CASE



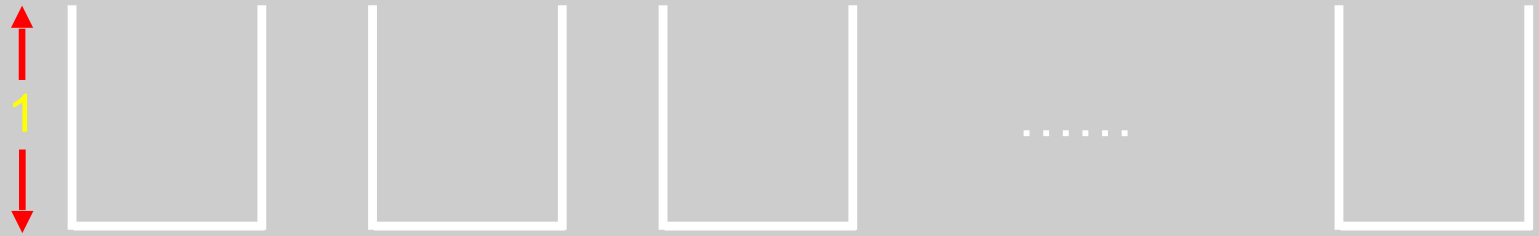
ENERGY USE CASE



Bin Packing Problem

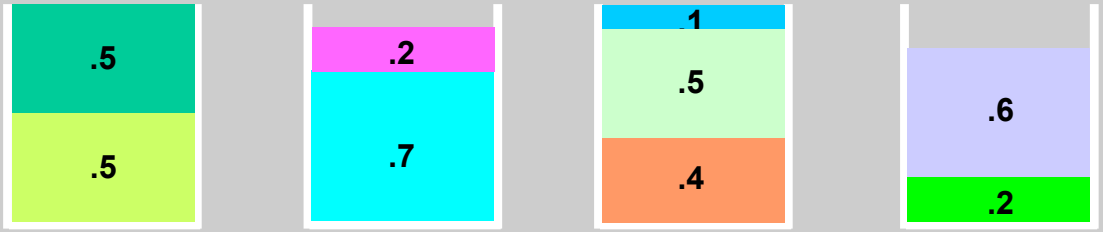


Bin Packing Problem



.5 .7 .5 .2 .4 .2 .5 .1 .6

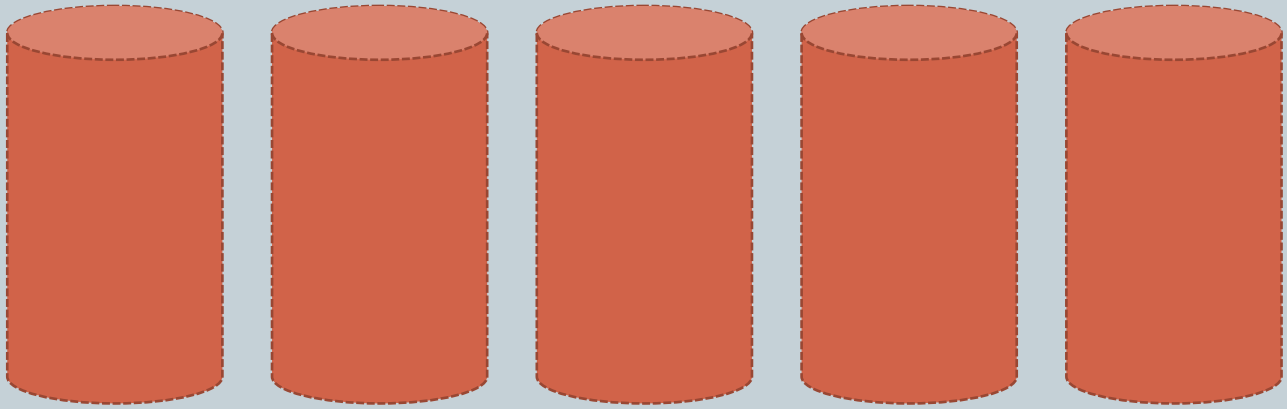
Optimal Packing



$N_0 = 4$

First fit algorithm

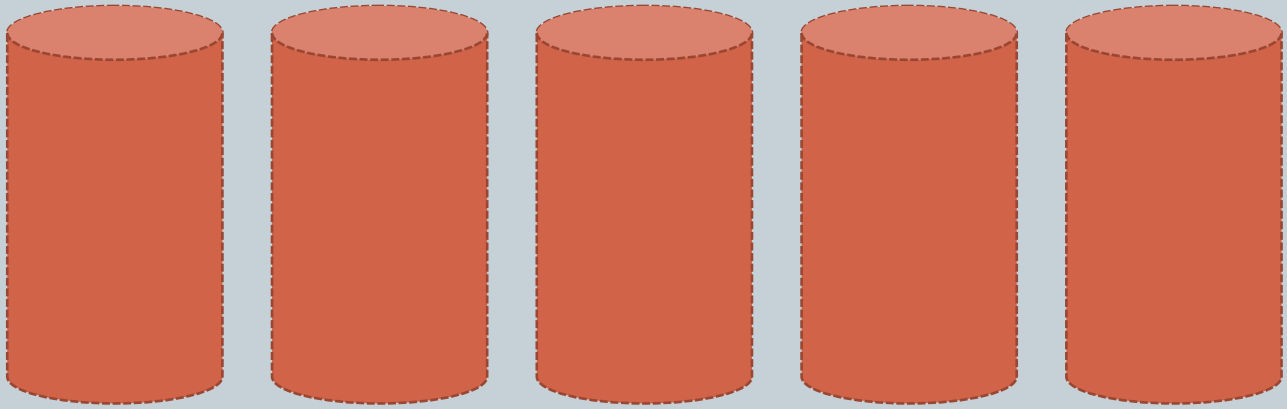
Bin size = 20



Bin 1 Bin 2 Bin 3 Bin 4 Bin 5

First fit algorithm

Bin size = 19

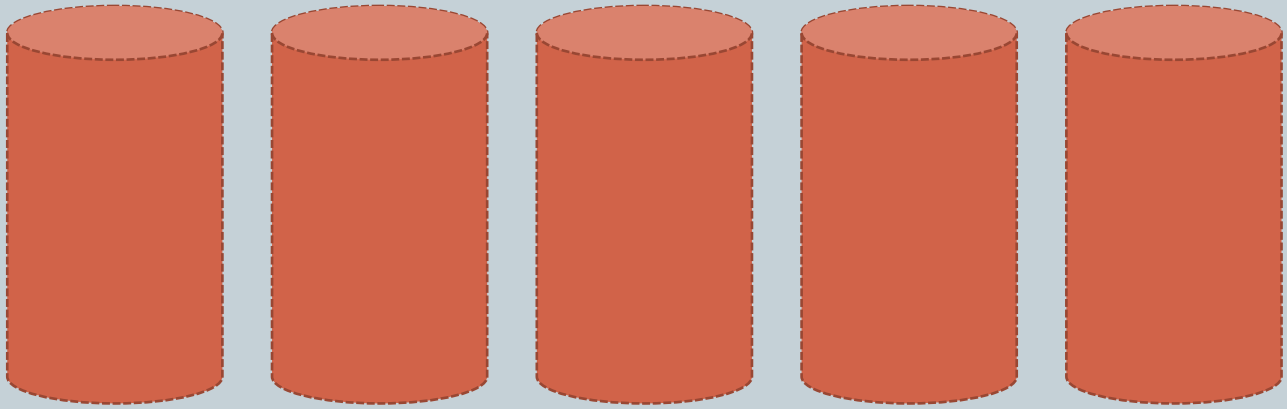


Bin 1 Bin 2 Bin 3 Bin 4 Bin 5

First fit decreasing algorithm

Sort values in size order, lowest first ...

Bin size = 20



Bin 1 Bin 2 Bin 3 Bin 4 Bin 5

- **Virtual Machine Placement Problem (VMPP) is similar to the multi-dimensional bin packing problem know to be NP-Hard ... [2007-02]**
- **Heuristic methods**
  - Greedy algorithms Ex: EnaCloud [2009-03]

    Construct a solution by taking local decision without backtrack.
    First-Fit Decrease (FFD), Best-Fit (BF), Worst-Fit (WF), Next-Fit (NF) ... [1997-01]
    Pro: Ease to implement, good worst-case complexity
    Cons: No optimal solution, not realy flexible
  - Metaheuristic Ex: Snooze [2012-04]

    Probailistic algorithms by searching near optimal solution
    Genetic, Tabu, Ant colony, Graps ...
    Pro: Better solution than Greedy algorithms
    Cons: No optimal solution, not realy flexible
- **Exact mehods**
  - Mathematical Ex: Entropy [2009-06]

    Linear or Constraint programming [1986-05]
    Compute optimal solution
    Pro: optimal and flexible
    Cons: Exponantial time solving process

MAIN CONCEPT FOR PPC

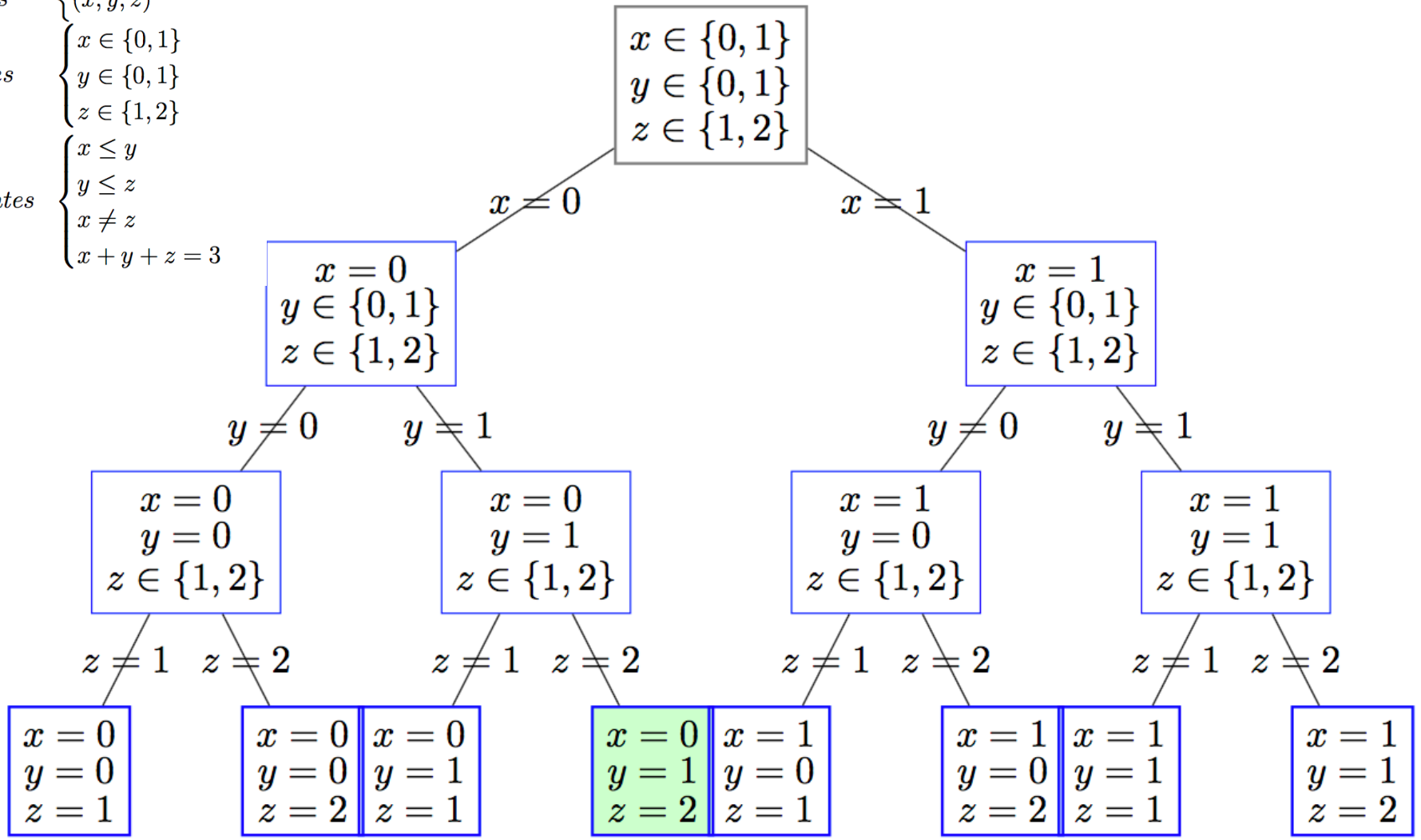
$$
\begin{cases}
\textbf{variables} & \{(x, y, z) \\
\textbf{domaines} & \begin{cases} x \in \{0, 1\} \\ y \in \{0, 1\} \\ z \in \{1, 2\} \end{cases} \\
\textbf{contraintes} & \begin{cases} x \leq y \\ y \leq z \\ x \neq z \\ x + y + z = 3 \end{cases}
\end{cases}
$$

SEARCH TREE

$$\text{variables} \quad \{(x, y, z)$$

$$\text{domaines} \quad \begin{cases} x \in \{0,1\} \\ y \in \{0,1\} \\ z \in \{1,2\} \end{cases}$$

$$\text{contraintes} \quad \begin{cases} x \leq y \\ y \leq z \\ x \neq z \\ x + y + z = 3 \end{cases}$$

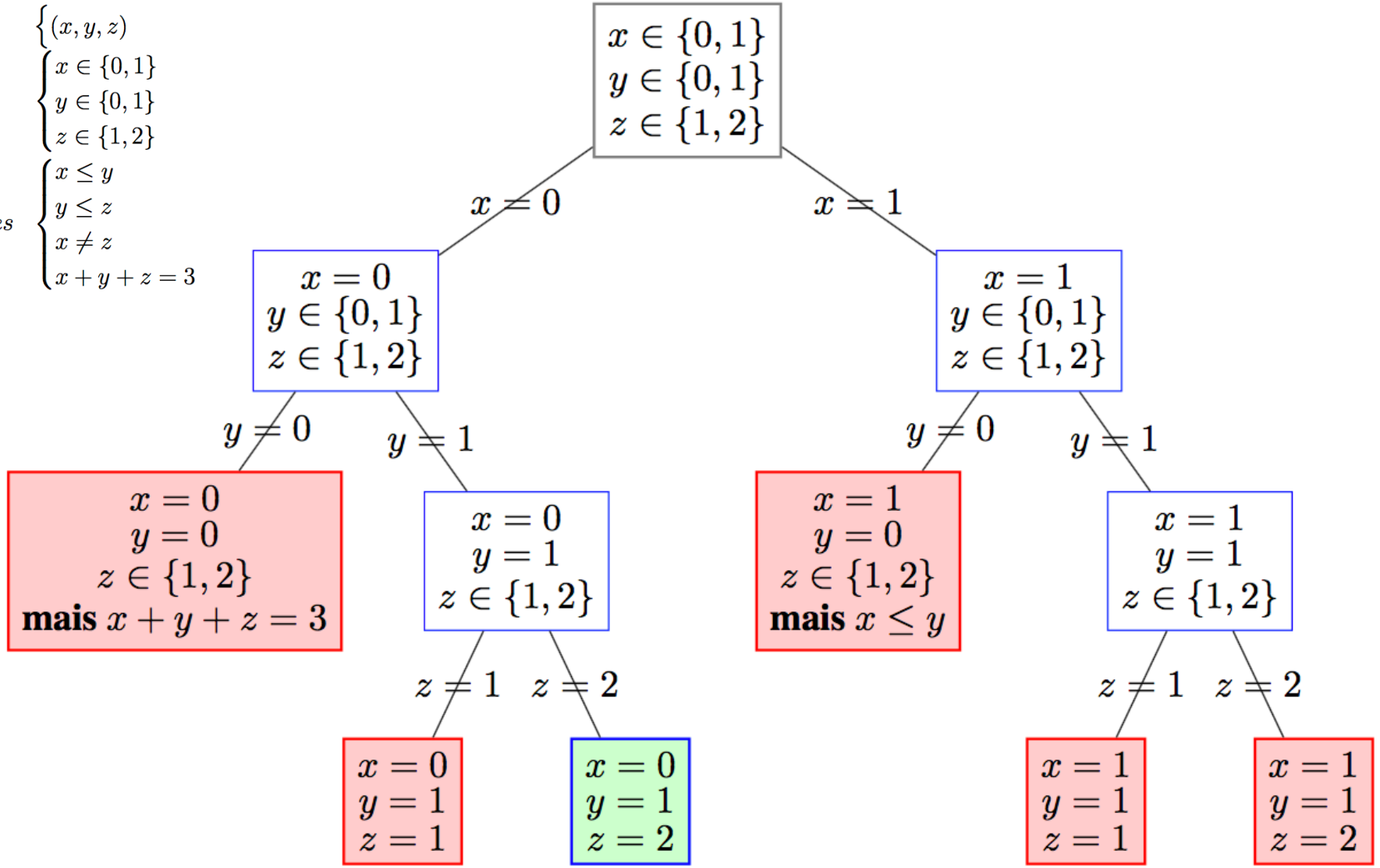


BACKTRACK

$$\text{variables} \quad \{(x, y, z)$$

$$\text{domaines} \quad \begin{cases} x \in \{0,1\} \\ y \in \{0,1\} \\ z \in \{1,2\} \end{cases}$$

$$\text{contraintes} \quad \begin{cases} x \leq y \\ y \leq z \\ x \neq z \\ x + y + z = 3 \end{cases}$$



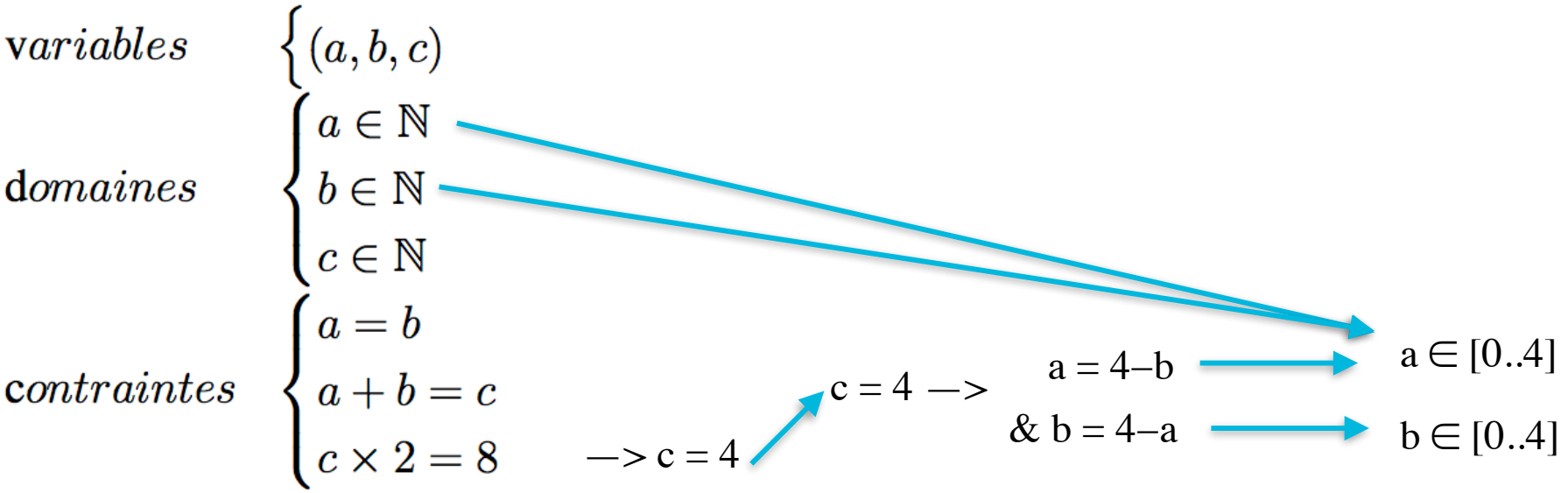
CONSTRAINT FILTER

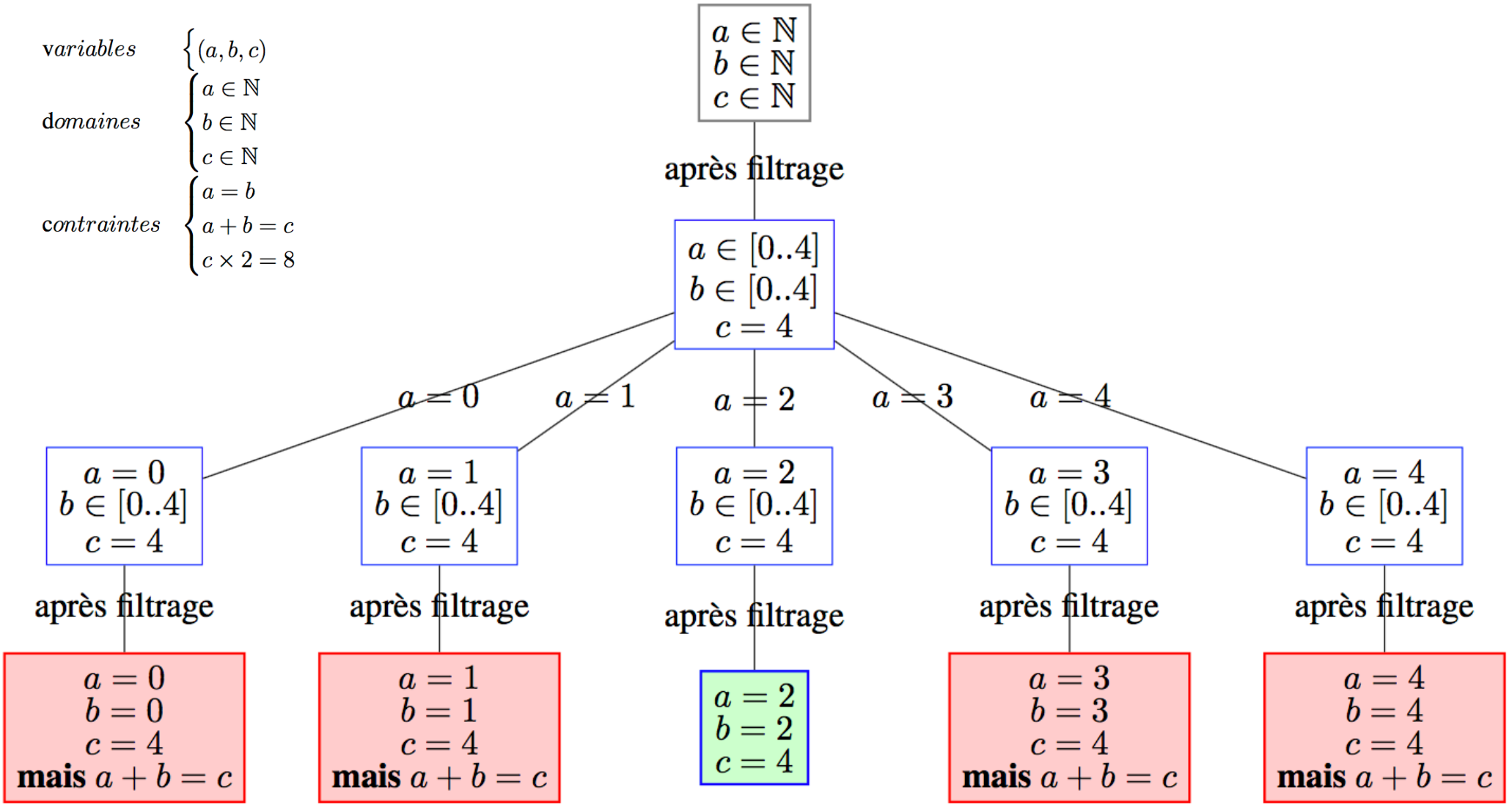
$$
\begin{array}{ll}
\textbf{v}ariables & \{(a,b,c) \\
\textbf{d}omaines & \begin{cases} a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \end{cases} \\
\textbf{c}ontraintes & \begin{cases} a = b \\ a + b = c \\ c \times 2 = 8 \end{cases}
\end{array}
$$

$$
\begin{array}{ll}
\textbf{variables} & \{(a, b, c) \\
\textbf{domaines} & \begin{cases} a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \end{cases} \\
\textbf{contraintes} & \begin{cases} a = b \\ a + b = c \\ c \times 2 = 8 \end{cases}
\end{array}
$$

c = 4 —> a = 4–b & b = 4–a

—> 2 × c = 8 —> c = 4


variables
(a, b, c)
domaines
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
contraintes
$a = b$
$a + b = c$
$c \times 2 = 8$
—> c = 4
c = 4 —>
a = 4–b
& b = 4–a
a ∈ [0..4]
b ∈ [0..4]


variables (a, b, c)
domaines $a \in \mathbb{N}$, $b \in \mathbb{N}$, $c \in \mathbb{N}$
contraintes $a = b$, $a + b = c$, $c \times 2 = 8$
$a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$
après filtrage
$a \in [0..4]$ $b \in [0..4]$ $c = 4$
$a = 0$
$a = 1$
$a = 2$
$a = 3$
$a = 4$
$a = 0$ $b \in [0..4]$ $c = 4$
$a = 1$ $b \in [0..4]$ $c = 4$
$a = 2$ $b \in [0..4]$ $c = 4$
$a = 3$ $b \in [0..4]$ $c = 4$
$a = 4$ $b \in [0..4]$ $c = 4$
après filtrage
après filtrage
après filtrage
après filtrage
après filtrage
$a = 0$ $b = 0$ $c = 4$ **mais** $a + b = c$
$a = 1$ $b = 1$ $c = 4$ **mais** $a + b = c$
$a = 2$ $b = 2$ $c = 4$
$a = 3$ $b = 3$ $c = 4$ **mais** $a + b = c$
$a = 4$ $b = 4$ $c = 4$ **mais** $a + b = c$

- **The approach : constraint programming**
  - generation of a core model
  - placement constraints are translated into "CP constraints"

$$
\begin{aligned}
\mathcal{X} &= \{x_1, x_2, x_3\} \\
\mathcal{D}(x_i) &= [0, 4], \forall x_i \in \mathcal{X} \\
\mathcal{C} &= \begin{cases} c_1 : & x_1 < x_2 \\ c_2 : & x_1 + x_2 + x_3 = 4 \\ c_3 : & allDifferent(x_1, x_2, x_3) \end{cases}
\end{aligned}
$$

CHAPTER 4.4
VIRTUALIZATION MARKET

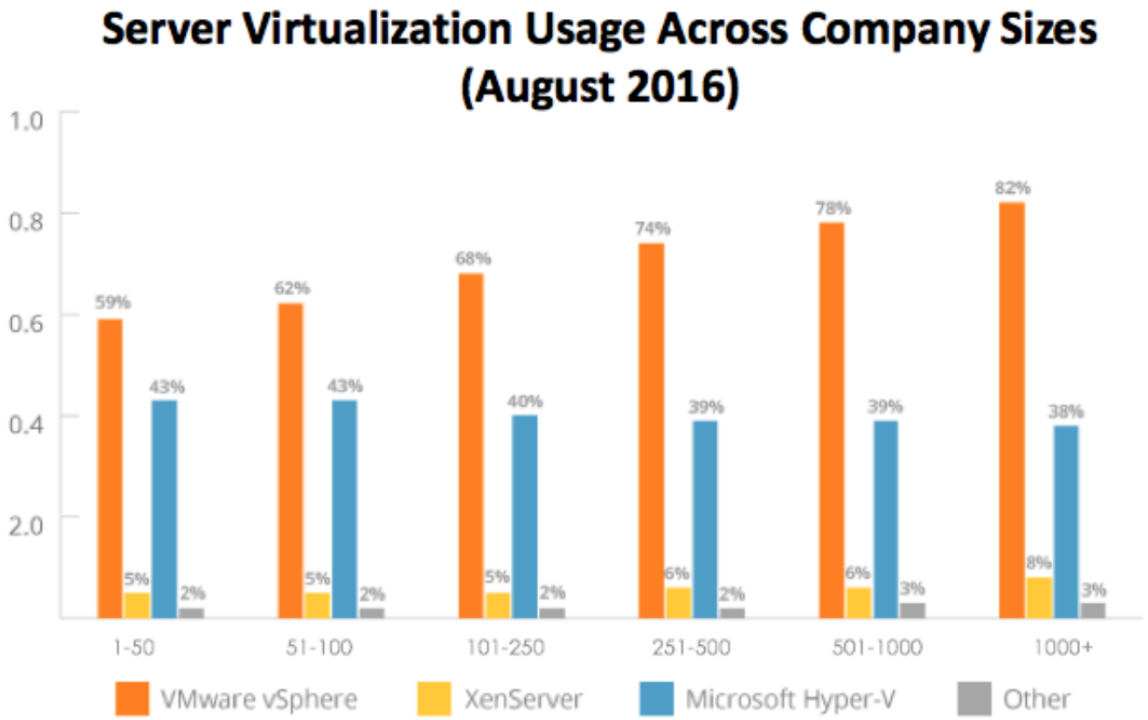
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

HYPERVISOR MARKET

Implementation	Technology
~~KQEMU [9]~~	~~Type II full virtualization~~
KVM [63]	Type II full virtualization
Lguest [72]	Type II para-virtualization
Linux-VServer [48]	OS-level virtualization
LXC [20]	OS-level virtualization
OpenVZ [83]	OS-level virtualization
QEMU [9]	Hardware emulation
Solaris Containers [80]	OS-level virtualization
Sun xVM Server [77]	Type I full virtualization with para-virtualization support
Sun xVM VirtualBox [76]	Type II full virtualization with para-virtualization support
~~UML [22]~~	~~Type II para-virtualization~~
~~UMLinux [39]~~	~~Type II para-virtualization~~
VMware ESX & ESXi [86, 88]	Type I full virtualization with para-virtualization support
VMware Workstation [89]	Type II full virtualization with para-virtualization support
Wind River Hypervisor [93]	Type II full virtualization with para-virtualization support
Xen [94]	Type I full- or para-virtualization

HYPERVISOR MARKET

Inspection by company size, server virtualization usage at large enterprises heavily favors VMware vSphere while smaller organizations use Hyper-V at a higher rate. For example, 42.6% of the smallest businesses utilize Hyper-V but only 37.9% of the largest companies do. In contrast, 58.6% of the smallest companies use the VMware vSphere ESXi hypervisor, compared to a whopping 82.9% of the biggest companies.

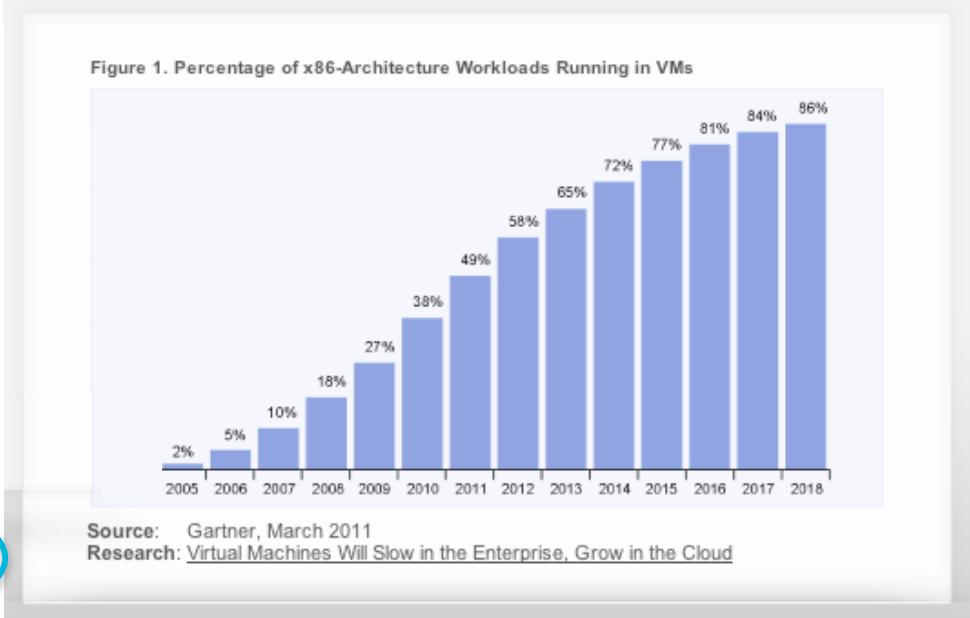


SPICEWORKS

VIRTUALIZATION MARKET

- **A constant progression**
- **Q3 2011 [2011-07]**
  - virtualization penetration rate: **38.9%**
  - Ratio of virtual machines to physical hosts: **5:1**
  - Primary Hypervisor Usage for Server virtualisation: **ESX 67,5%**

	Avg	UK	FR	DE	US
VMware	67.6	79	65	61	66.5
Citrix	14.4	12	15	20	12.5
Hyper-V	16.4	8	17	16	20.5
Other	1.6	1	3	3	0.5



Gartner March 2011

The Periodic Table of the VMware Solutions

delivery / lic	Perpetual	Subscription
On-Premises	Perpetual	Term
Cloud	-	SaaS

SDDC & HCI | Cloud MGMT | Cloud & Telco | Networking | Modern Apps | Workspace | Security | Free / OpenSrc | Services

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Period 1	1 / P vSph vSphere																	2 PS Professional Services
2	3 / P vCent vCenter Server	4 / P / T vSAN vSAN											5 / P / T SDFW NSX Service-Defined FW	6 VEBA VMware Event Broker Applian.	7 Photon Photon OS	8 Flings VMware Flings	9 360 VMware Success 360	10 TAM Technical Acc Management
3	11 / P / T / S NSX NSX Data Center	12 / P / T / S vROPs vRealize Operations											13 / P / T IDS/IPS NSX Distrib. IDS/IPS	14 Antrea Antrea	15 Harbor Harbor	16 Velero Velero	17 Learn Customer Con. Learning	18 Cert Certification
4	19 / P / T / S vRLI vRealize Log Insight	20 / P / T vRLCM vRealize Lifecycle Mgr	21 / S vR AI vRealize AI Cloud	22 / P / T / S vRS vRealize Suite	23 / S VMCoA VMware Cloud on AWS	24 / T VMCoD VMware Cloud on Dell EMC	25 / S VMCU VMware Cloud Universal	26 / S AVS Azure VMware Solution	27 / S GCVE Google Cloud VMware Eng.	28 / S IBMVM IBM Cloud for VMware Sol.	29 / S OCVS Oracle Cloud VMware Sol.	30 / S ACVS Alibaba Cloud VMware Sol.	31 / T NDR NSX Detection & Response	32 / T ATP NSX Adv Threat Prevention	33 Cont Contour	34 Sonob Sonobuoy	35 SnS Support and Services	36 CustC Customer Connect
5	37 / P / T / S vRA vRealize Automation	38 / P / T vRO vRealize Orchestrator	39 / S CH Cloud Health	40 / P / T vCS vCloud Suite	41 / P VCF VMware Cloud Foundation	42 / P SRM Site Recovery Manager	43 / S CDR Cloud Disaster Recovery	44 5G 5G	45 TCP Telco Cloud Platform	46 TCA Telco Cloud Automation	47 TCOps Telco Cloud Operations	48 TCI Telco Cloud Infrastructure	49 / S CBE Carbon Black Endpoint	50 / T / S EDR Carbon Black EDR	51 Carvel Carvel	52 Pinni Pinniped	53 TCE Tanzu Community Ed.	54 Sky VMware Skyline
6	55 / P / T / S vRNI vRealize Net Insight	56 / P TVS True Visibility Suite	57-71 Modern Apps	72 / S vRCU vRealize Cloud Universal	73 CD Cloud Director	74 VCP Cloud Provider Platform	75 / S VMM VMware Marketplace	76 / P / T / S WS1 Workspace ONE	77 / P / T / S UEM Workspace ONE UEM	78 / P / T / S Intelli Worksp ONE Intelligence	79 / P / T / S Assist Workspace ONE Assist	80 / P / T / S Access Workspace ONE Access	81 / P / T / S HUB Worksp ONE Intelligent Hub	82 / T CBApp Carbon Black App Control	83 Clarity Clarity Design System	84 Spring Spring Projects	85 Herald Herald	86 Octant Octant
7	87 / P SSC SaltStack Config	88 / P SSCO SaltStack SecOps	89-103 Modern Apps	104 / P / T / S NSXALB NSX Advanced Load Balancer	105 / S NSXC NSX Cloud	106 / P / T / S SDWAN NSX SD-WAN	107 / P / T / S HCX HCX	108 / P / T NSXInt NSX Intelligence	109 / T / S HZ Horizon	110 / S HZC Horizon Cloud	110 / T / S HZApps Horizon Apps	112 / T AppV App Volumes	113 / T DEM Dynamic Env Manager	114 / P WKS/F Desktop Hypervisor	115 / S SASE Secure Access Service Edge	116 Conv vCenter Converter	117 ESXi vSphere Hypervisor	118 PCLI PowerCLI
Modern Apps				57 / T TanzuB Tanzu Basic	58 / T TzSTD Tanzu Standard	59 / T TzADV Tanzu Advanced	60 / T TKGs TZ Kubernetes Grid Service	61 / T / S TKGI TZ Kubernetes Grid Integrated	62 / T / S TKG TZ Kubernetes Grid	63 / S TMC Tanzu Mission Control	64 / S TO Tanzu Observability	65 / S TSM Tanzu Service Mesh	66 / T TBS Tanzu Build Service	67 / T TAS Tanzu App Service	68 / T TBP Tanzu Bulidpacks	69 / S VAC VMware App Catalog	70 / T TDS Tanzu Data Services	71 / T GeF GemFire
				89 / T RabMQ RabbitMQ	90 / T SQL Tanzu SQL	91 / T Post VMware Postgres	92 / T GreP Greenplum	93 / T ALBe NSX (AVI) ADV LB Essentials	94 / T ConC Concourse	95 / T TAP Tanzu App Platform	96 Spring Spring	97 SprB Spring Boot	98 SprCld Spring Cloud	99 SprCDF Spring Cloud Data Flow	100 SprSec Spring Security	101 / S AzSpr Azure Spring Cloud	102 TzOps Tanzu Ops Manager	103 TzTool Tanzu Toolkit for Kubernetes

vmware

https://tanzu.vmware.com/
https://www.vmware.com/
https://vmware.github.io/
https://docs.vmware.com/

v1.1, 16th of Feb, 2022 aherceg@vmware.com for the latest version: https://vthing.wordpress.com/

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

The Periodic Table of the VMware Solutions

delivery / lic	Perpetual	Subscription
On-Premises	Perpetual	Term
Cloud	-	SaaS

SDDC & HCI | Cloud MGMT | Cloud & Telco | Networking | Modern Apps | Workspace | Security | Free / OpenSrc | Services

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Period 1	1 / P vSph vSphere																	2 PS Professional Services
Period 2	3 / P vCent vCenter Server	/ P / T vSAN vSAN											5 / P / T SDFW NSX Service-Defined FW	6 VEBA VMware Event Broker Applian.	7 Photon Photon OS	8 Flings VMware Flings	9 360 VMware Success 360	10 TAM Technical Acc Management
Period 3	11 / P / T / S NSX NSX Data Center	12 / P / T / S vROPs vRealize Operations											13 / P / T IDS/IPS NSX Distrib. IDS/IPS	14 Antrea Antrea	15 Harbor Harbor	16 Velero Velero	17 Learn Customer Con. Learning	18 Cert Certification
Period 4	19 / P / T / S vRLI vRealize Log Insight	20 / P / T vRLCM vRealize Lifecycle Mgr	21 / S vR AI vRealize AI Cloud	22 / P / T / S vRS vRealize Suite	23 / S VMCoA VMware Cloud on AWS	24 / T VMCoD VMware Cloud on Dell EMC	25 / S VMCU VMware Cloud Universal	26 / S AVS Azure VMware Solution	27 / S GCVE Google Cloud VMware Eng.	28 / S IBMVM IBM Cloud for VMware Sol.	29 / S OCVS Oracle Cloud VMware Sol.	30 / S ACVS Alibaba Cloud VMware Sol.	31 / T NDR NSX Detection & Response	32 / T ATP NSX Adv Threat Prevention	33 Cont Contour	34 Sonob Sonobuoy	35 SnS Support and Services	36 CustC Customer Connect
Period 5	37 / P / T / S vRA vRealize Automation	38 / P / T vRO vRealize Orchestrator	39 / S CH Cloud Health	40 / P / T vCS vCloud Suite	41 / P VCF VMware Cloud Foundation	42 / P SRM Site Recovery Manager	43 / S CDR Cloud Disaster Recovery	44 5G 5G	45 TCP Telco Cloud Platform	46 TCA Telco Cloud Automation	47 TCOps Telco Cloud Operations	48 TCI Telco Cloud Infrastructure	49 / S CBE Carbon Black Endpoint	50 / T / S EDR Carbon Black EDR	51 Carvel Carvel	52 Pinni Pinniped	53 TCE Tanzu Community Ed.	54 Sky VMware Skyline
Period 6	55 / P / T / S vRNI vRealize Net Insight	56 / P TVS True Visibility Suite	57-71 Modern Apps	72 / S vRCU vRealize Cloud Universal	73 CD Cloud Director	74 VCP Cloud Provider Platform	75 / S VMM VMware Marketplace	76 / P / T / S WS1 Workspace ONE	77 / P / T / S UEM Workspace ONE UEM	78 / P / T / S Intelli Worksp ONE Intelligence	79 / P / T / S Assist Workspace ONE Assist	80 / P / T / S Access Workspace ONE Access	81 / P / T / S HUB Worksp ONE Intelligent Hub	82 / T CBApp Carbon Black App Control	83 Clarity Clarity Design System	84 Spring Spring Projects	85 Herald Herald	86 Octant Octant
Period 7	87 / P SSC SaltStack Config	88 / P SSCO SaltStack SecOps	89-103 Modern Apps	104 / P / T / S NSXALB NSX Advanced Load Balancer	105 / S NSXC NSX Cloud	106 / P / T / S SDWAN NSX SD-WAN	107 / P / T / S HCX HCX	108 / P / T NSXInt NSX Intelligence	109 / T / S HZ Horizon	110 / S HZC Horizon Cloud	110 / T / S HZApps Horizon Apps	112 / T AppV App Volumes	113 / T DEM Dynamic Env Manager	114 / P WKS/F Desktop Hypervisor	115 / S SASE Secure Access Service Edge	116 Conv vCenter Converter	117 ESXi vSphere Hypervisor	118 PCLI PowerCLI
Modern Apps				57 / T TanzuB Tanzu Basic	58 / T TzSTD Tanzu Standard	59 / T TzADV Tanzu Advanced	60 / T TKGs TZ Kubernetes Grid Service	61 / T / S TKGI TZ Kubernetes Grid Integrated	62 / T / S TKG TZ Kubernetes Grid	63 / S TMC Tanzu Mission Control	64 / S TO Tanzu Observability	65 / S TSM Tanzu Service Mesh	66 / T TBS Tanzu Build Service	67 / T TAS Tanzu App Service	68 / T TBP Tanzu Bulidpacks	69 / S VAC VMware App Catalog	70 / T TDS Tanzu Data Services	71 / T GeF GemFire
				89 / T RabMQ RabbitMQ	90 / T SQL Tanzu SQL	91 / T Post VMware Postgres	92 / T GreP Greenplum	93 / T ALBe NSX (AVI) ADV LB Essentials	94 / T ConC Concourse	95 / T TAP Tanzu App Platform	96 Spring Spring	97 SprB Spring Boot	98 SprCld Spring Cloud	99 SprCDF Spring Cloud Data Flow	100 SprSec Spring Security	101 / S AzSpr Azure Spring Cloud	102 TzOps Tanzu Ops Manager	103 TzTool Tanzu Toolkit for Kubernetes

vmware

https://tanzu.vmware.com/
https://www.vmware.com/
https://vmware.github.io/
https://docs.vmware.com/

v1.1, 16th of Feb, 2022 aherceg@vmware.com for the latest version: https://vthing.wordpress.com/

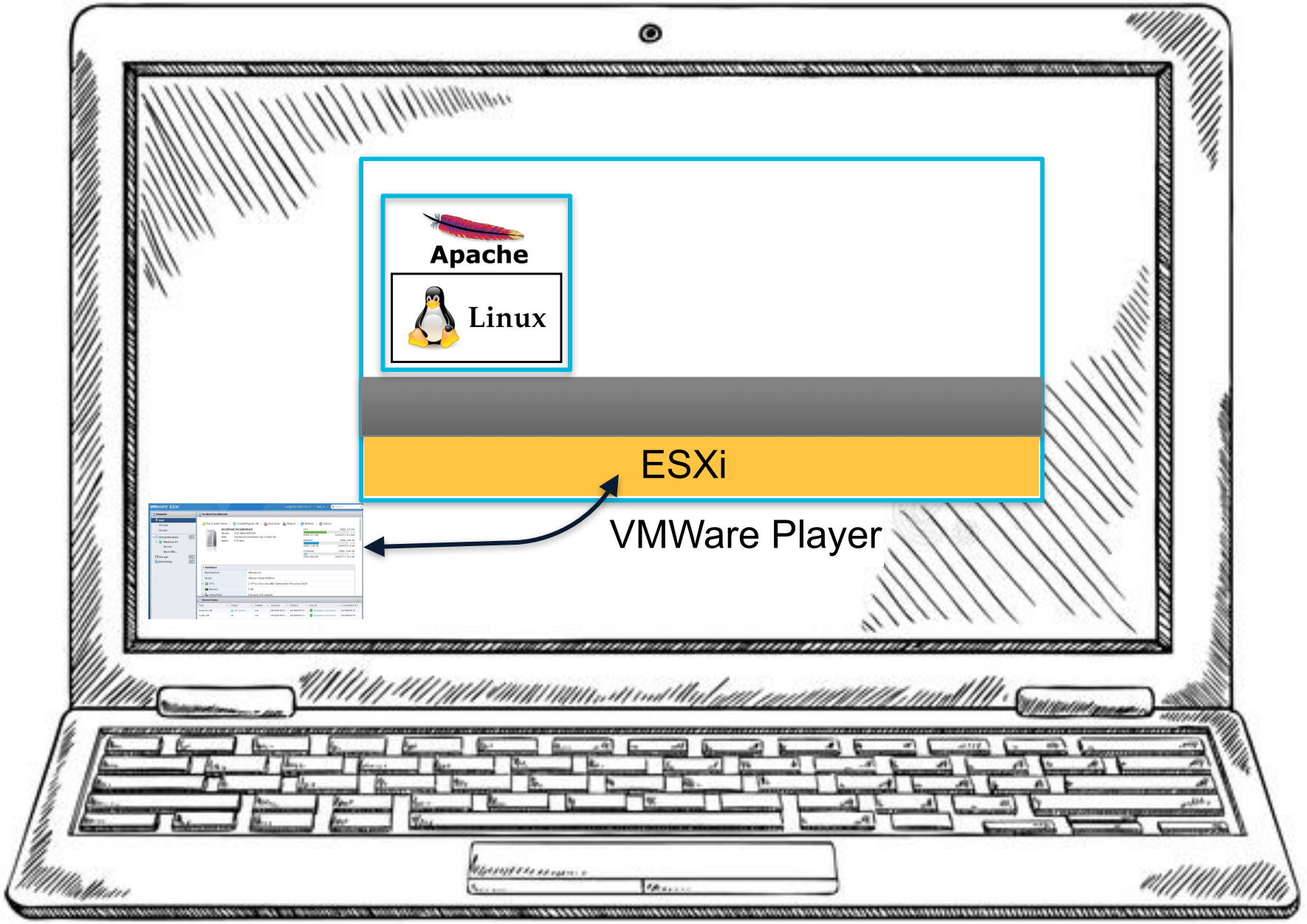
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CHAPTER 4.4
VIRTUALIZATION
PRACTICAL SESSION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

PRACTICAL SESSION 1

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom


Apache
Linux
ESXi
VMWare Player

ESXi 6.5 Installation

What you need :

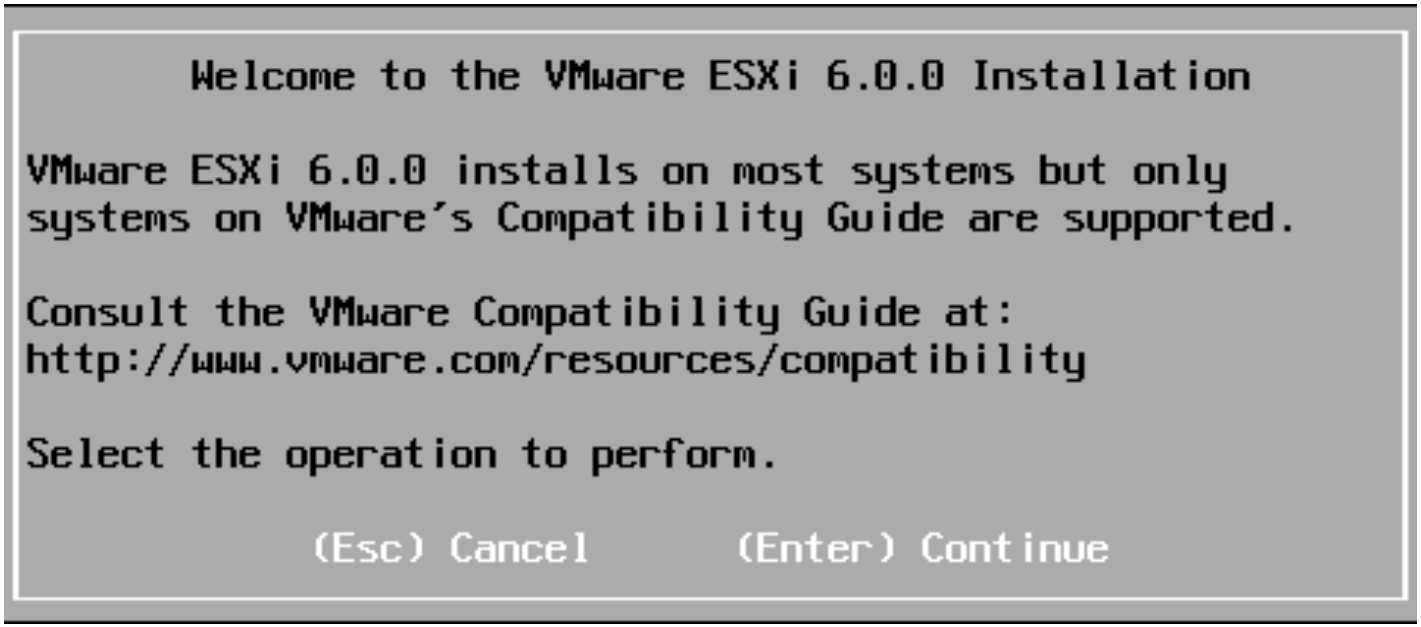
Hardware (on your laptop):

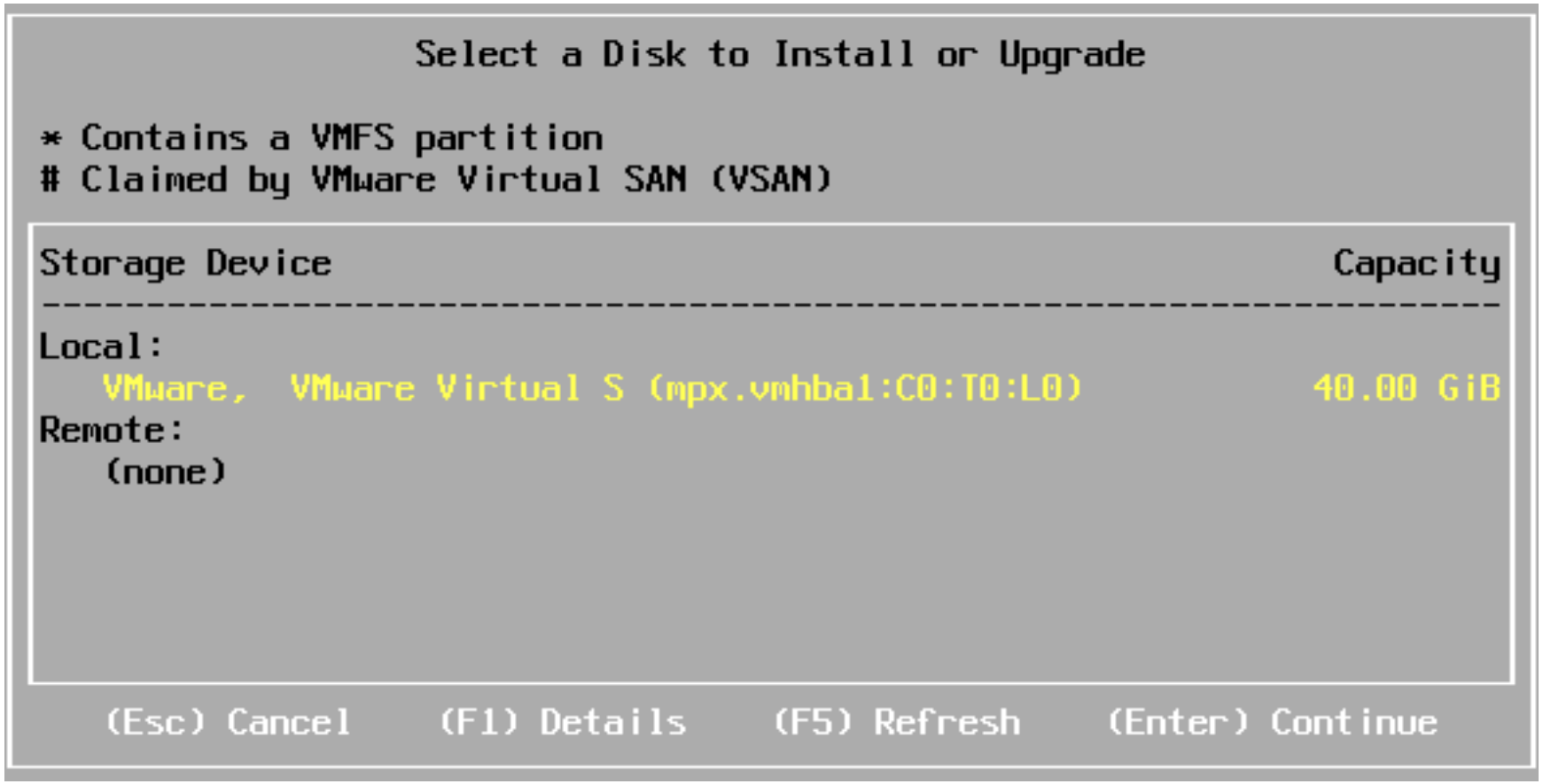
- **4 Go of RAM**
- **40Go of disk**
- **A Network connexion**

Software

- **VMWare Player**
- **ESXi Iso**
- **DOS.ova**
- **TinyLinux.ova**

ESXI INSTALLATION

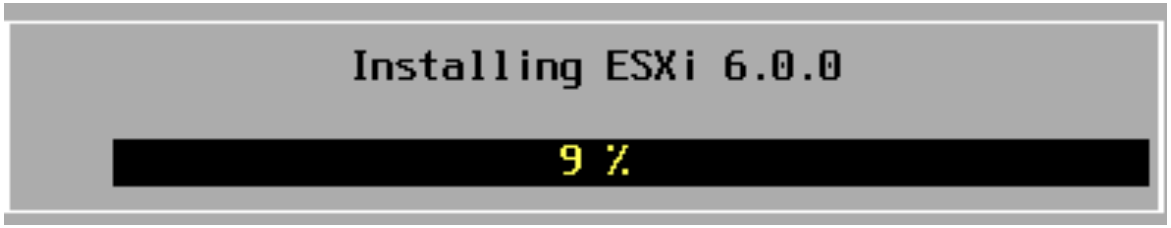



Select a Disk to Install or Upgrade
* Contains a VMFS partition
# Claimed by VMware Virtual SAN (VSAN)
Storage Device
Capacity
Local:
VMware, VMware Virtual S (mpx.vmhba1:C0:T0:L0)
40.00 GiB
Remote:
(none)
(Esc) Cancel
(F1) Details
(F5) Refresh
(Enter) Continue

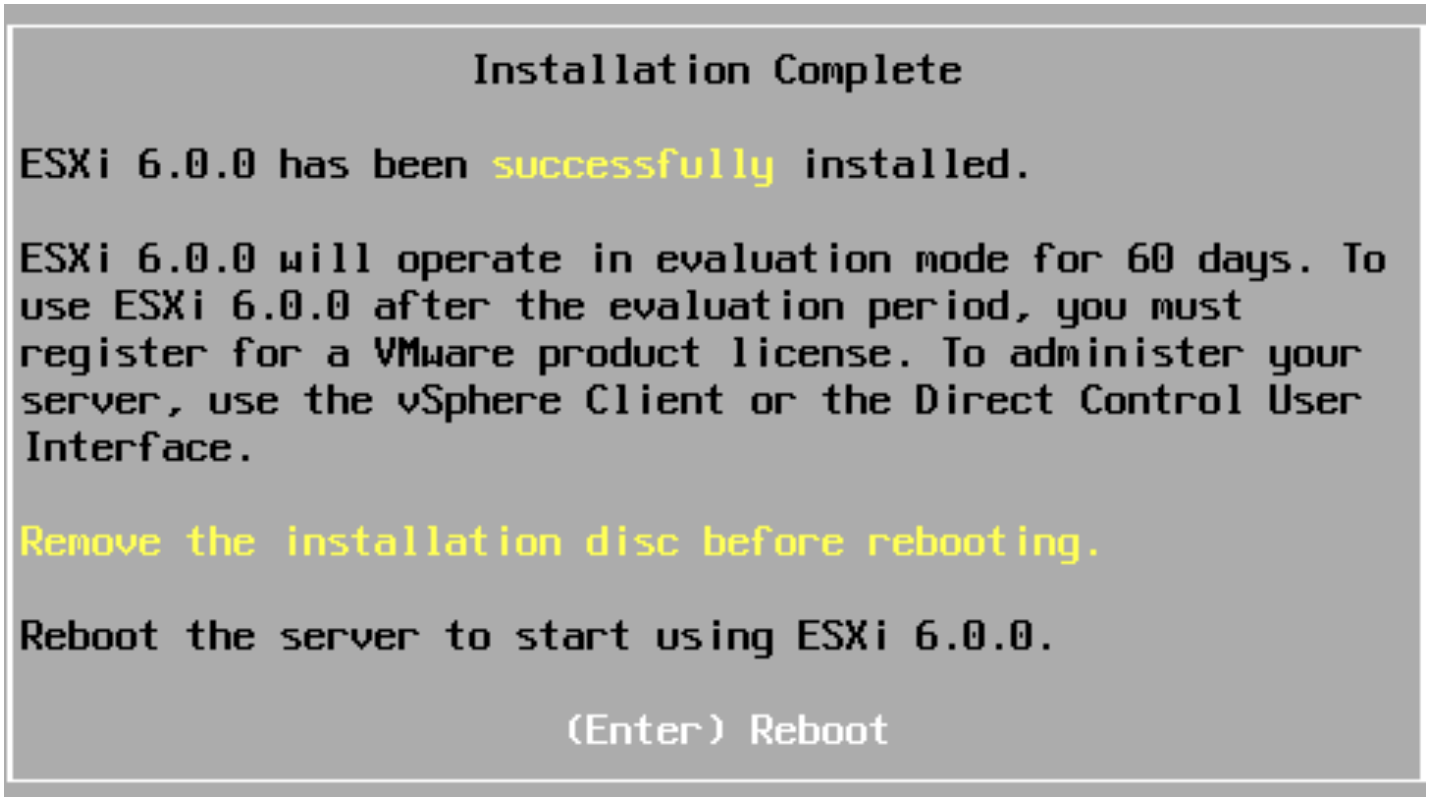
Please, don’t forget your password !!!!

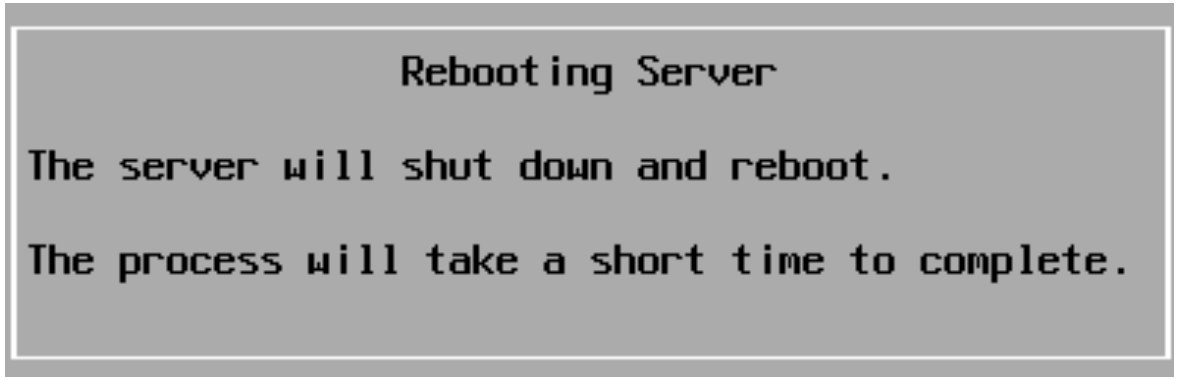
ESXI INSTALLATION

ESXI INSTALLATION



ESXI INSTALLATION

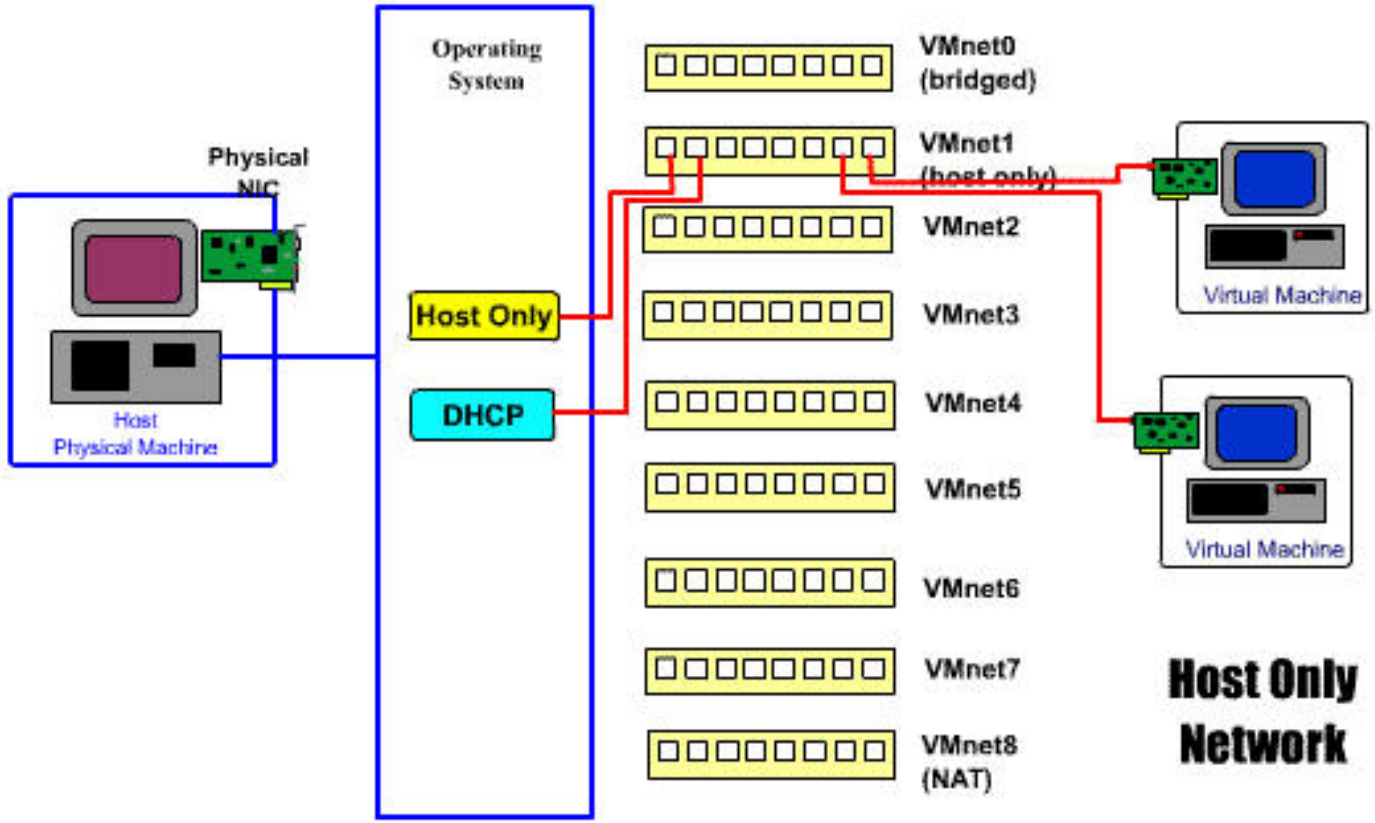



Rebooting Server
The server will shut down and reboot.
The process will take a short time to complete.

ESXI INSTALLATION

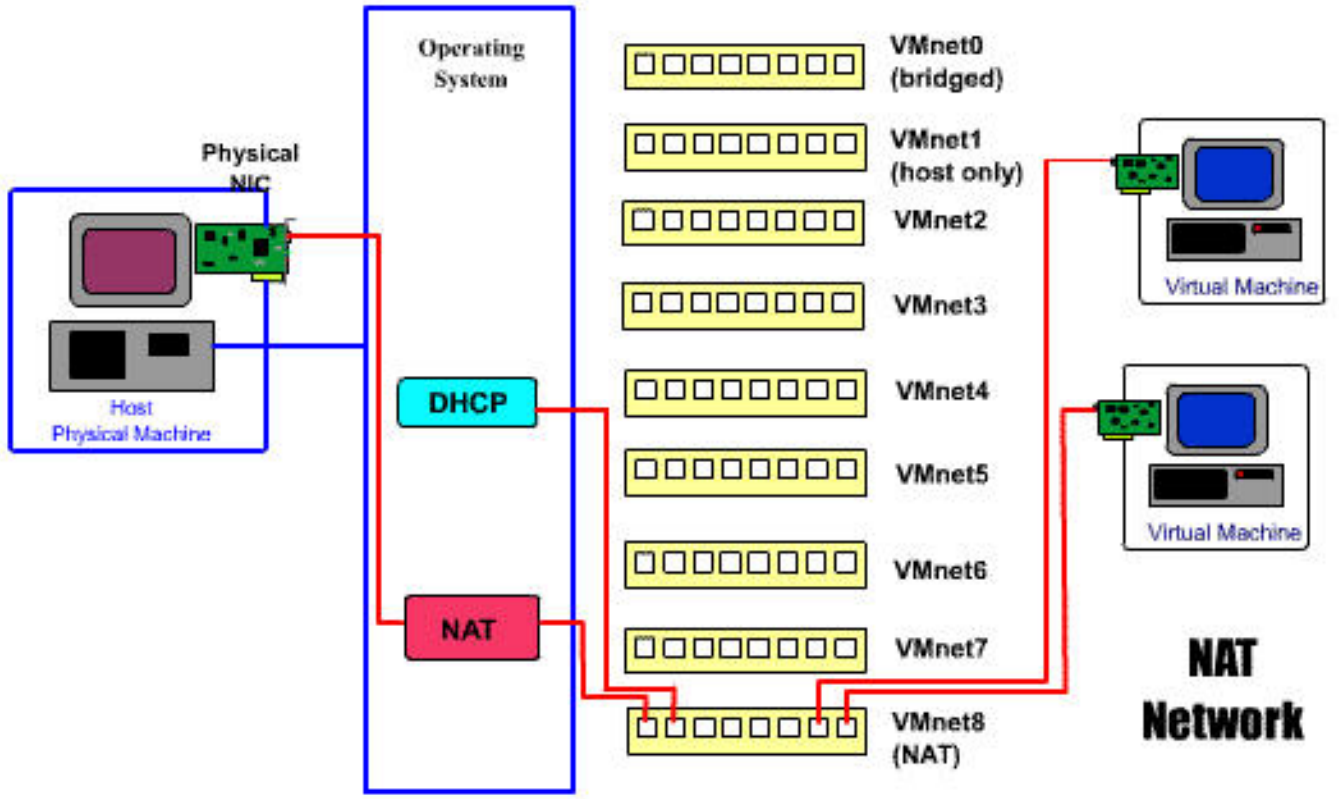
HOST, NAT, BRIDGE ????

- **Host-Only**: The VM will be assigned one IP, but it's only accessible by the box VM is running on. No other computers can access it.



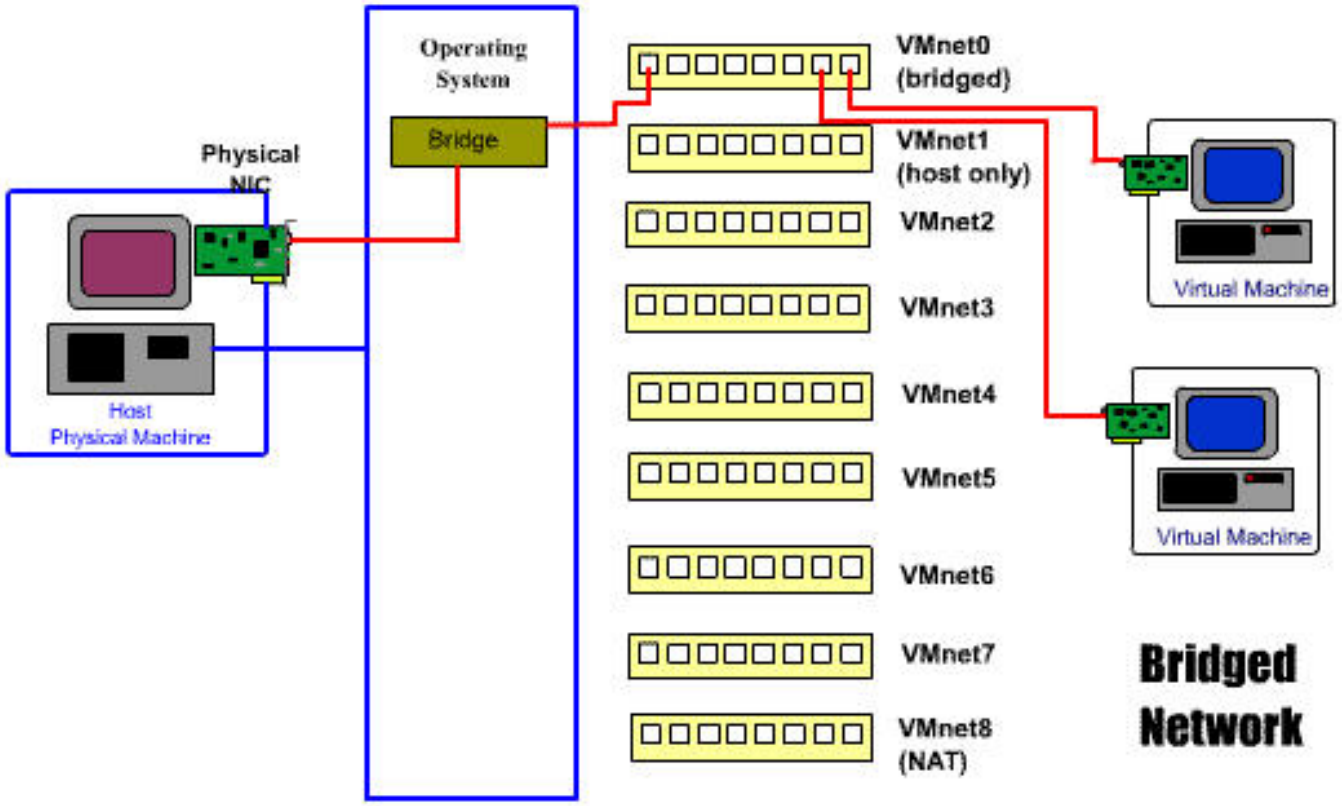
HOST, NAT, BRIDGE ????

- **NAT**: Just like your home network with a wireless router, the VM will be assigned in a separate subnet, like 192.168.6.1 is your host computer, and VM is 192.168.6.3, the your VM can access outside network like your host, but no outside access to your VM directly, it's protected.



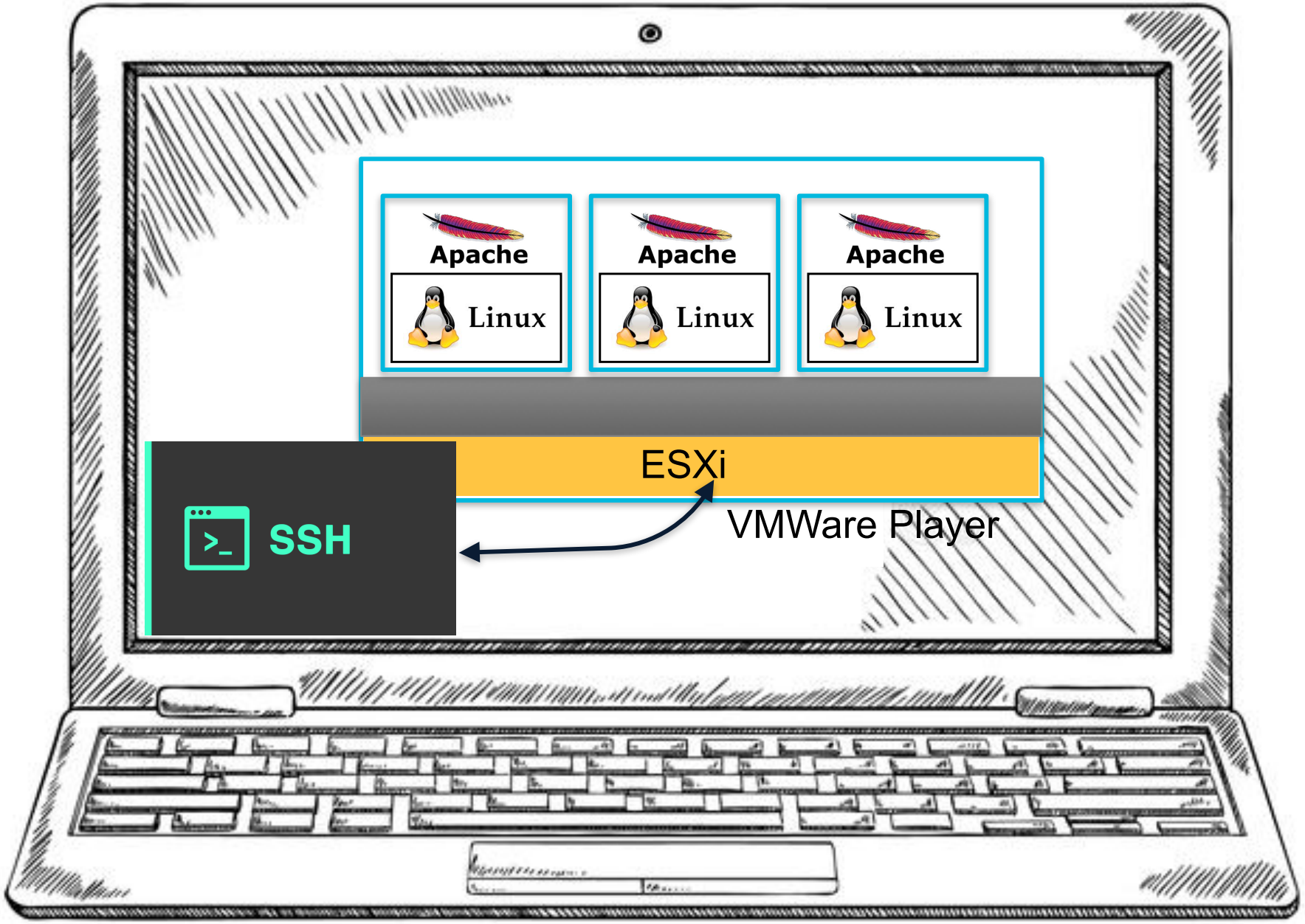
HOST, NAT, BRIDGE ????

- **Bridged**: Your VM will be in the same network as your host, if your host IP is 172.16.120.45 then your VM will be like 172.16.120.50. It can be accessed by all computers in your host network.



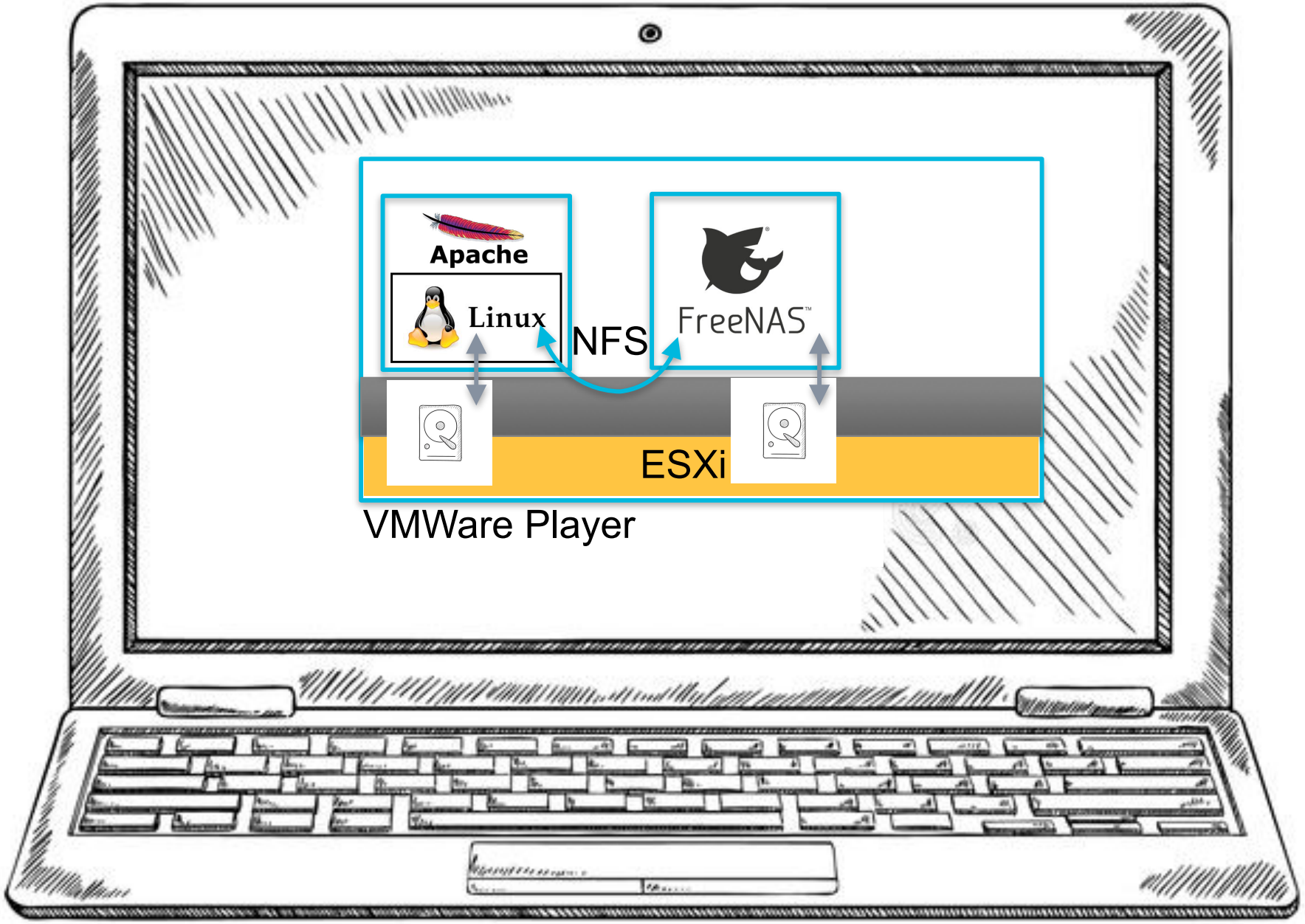
PRACTICAL SESSION 2

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom


Apache
Linux
Apache
Linux
Apache
Linux
ESXi
VMWare Player
SSH

PRACTICAL SESSION 3

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom


Apache
Linux
NFS
FreeNAS
ESXi
VMWare Player

PRACTICAL SESSION 4

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

WEB SITE

https://ecadorel.gitlabpages.inria.fr/ChocLet_release/

```
> let x = 10
0ms > let y ="totto totot"
0ms > y
totto totot
0ms > x
10
0ms > let z = 3.415
3ms > z
3.415
2ms > x = 2.2
ast.exception.BinaryException:
-----------------------------------------
Error : undefined op '=' for types 'int' and 'double'
 --> !(1,3)
    |
 1  | x = 2.2
    |   ^
-----------------------------------------
```

```
> let x = [1,2,3]
1ms > x
[1, 2, 3]

    let y = [100 of float]
    1ms > y
    [null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null]

        let x = [1,2,3]~[3,4,5]
        2ms > x
        [1, 2, 3, 3, 4, 5]
         > println(x)
        [1, 2, 3, 3, 4, 5]
```

```
 > def foo(a,b){println(a,b);}
2ms > foo(3,4)
34
```

```
> let a = choco.int(0,10)
74ms > let b=choco.int(0,10)
0ms > a
IV_1 = {0,1,2,3,4,5...,10}
1ms > b
IV_2 = {0,1,2,3,4,5...,10}
```

```
(a!=b).post()
10ms > choco.print()
Model[Model-0]

[ 3 vars -- 2 cstrs ]
satisfaction  : true
== variables ==
IV_1 = 1
IV_2 = 0
NE_exp_3 = 1
== constraints ==
BASIC_REIF ([(IV_1 = IV_2) <=> NE_exp_3])
ARITHM ([NE_exp_3 = 1])
```

```
println(a,b)
IV_1 = {0,1,2,3,4,5...,10}IV_2 = {0,1,2,3,4,5...,10}
0ms > choco.solve()
true
0ms > println(a,' ',b)
1 0
```

QUESTION 1

We are a cloud provider. We sold 5 VMs to ours clients.
Each VM has a ***cost*** (between 1 to 10 dollars) for our client.
print on the screen our profit ?

QUESTION 1

We are a cloud provider. We sold 5 VMs to ours clients.
Each VM has a ***cost*** (between 1 to 10 dollars) for our client.
print on the screen our profit ?

```
let VmCost = [10,3,5,1,9];

let c =0;

for i in 0..VmCost.len
 c = c + VmCost[i];

choco.solve();
println(c);
```

QUESTION 2

We are now a cloud provider !!!

We have ***5*** VM.
Each VM has a ***State*** (0=Off, 1=On) and a ***cost*** (between 1 to 10 dollars) for our client.
Our data center is limited, so we can't host all VM's clients.
In our new problem, we want exactly ***nb*** VM On.
print all VM States.

QUESTION 2

We are now a cloud provider !!!

We have ***5*** VM.
Each VM has a ***State*** (0=Off, 1=On) and a ***cost*** (between 1 to 10 dollars) for our client.
Our data center is limited, so we can't host all VM's clients.
In our new problem, we want exactly ***nb*** VM On.
print all VM States.

```
let VmCost = [10,3,5,1,9];
let VmState = choco.intArray("S",5,0,1);
let c = 3;


in choco {
 def sum -> ChocoConstraint;
}


choco.sum(VmState,'=',c).post();

choco.solve();
println(VmState);
```

QUESTION 3

We want exactly ***nbOn*** VM On and **maximize our profit**,

profit = the **sum** of VM on costs

print our profit

QUESTION 3

We want exactly ***nbOn*** VM On and **maximize our profit**,
profit = the **sum** of VM on costs
print our profit

```
let VmPrio = [10,3,5,1,9];
let VmState = choco.intArray("S",5,0,1);
let On = [5 of ChocoInt];
let c = 3;
let result = choco.int("R",0,1000);

in choco {
 def sum -> ChocoConstraint;
}

choco.sum(VmState,'=',c).post();


for it in 0..5 {
  On[it] = VmState[it]*VmPrio[it];
}

choco.sum(On,'=',result).post();

result.maximize();

while(choco.solve())
 println(VmState,result);
```

QUESTION 4

We add servers. Each server has a capacity. Each VM a consumption.
Each VM on, must be placed on a server.
We want exactly ***nbOn*** VM On and **maximize our profit**,

QUESTION 4

We add servers. Each server has a capacity. Each VM a consumption.
Each VM on, must be placed on a server.
We want exactly ***nbOn*** VM On and **maximize our profit**,

```
let VmPrio = [10,3,5,1,9];
let VmState = choco.intArray("S",5,0,1);
let On = [5 of ChocoInt];
let c = 3;
let result = choco.int("R",0,1000);

let VmCapa = [10,5,3,6,7];
let Server = [10,20];


in choco {
 def sum -> ChocoConstraint;
 def binPacking -> ChocoConstraint;
}
```

```
choco.sum(VmState,'=',c).post();

for it in 0..5 {
  On[it] = VmState[it]*VmPrio[it];
}

choco.sum(On,'=',result).post();

let tab = choco.intArray("T",5,0,2);

let binload =
[choco.int(0,Server[0]),choco.int(0,Server[1]),choco
.int(0,1000)];

choco.binPacking(tab,VmCapa,binload,0).post();

for it in 0..5 {
  (VmState[it] == 0) <-> (tab[it]==2);
}

result.maximize();

while(choco.solve())
 println(VmState,result,tab);
```

QUESTION 5

Each server has a VM power consumption, for each VM placed on it, the server consume Xw.
We want exactly ***nbOn*** VM On and minimize the **total power consumption**

QUESTION 5

Each server has a VM power consumption, for each VM placed on it, the server consume Xw.
We want exactly ***nbOn*** VM On and minimize the **total power consumption**

```
let VmPrio = [10,3,5,1,9];
let VmState = choco.intArray("S",5,0,1);
let On = [5 of ChocoInt];
let c = 3;
let result = choco.int("R",0,1000);

let VmCapa = [10,5,3,6,7];
let Server = [15,30];

let ServerPower = to![ChocoInt]([200,100,0]);
let power = choco.int("P",0);


in choco {
 def sum -> ChocoConstraint;
 def binPacking -> ChocoConstraint;
}
```

```
choco.sum(VmState,'=',c).post();

for it in 0..5 {
  On[it] = VmState[it]*VmPrio[it];
}

choco.sum(On,'=',result).post();

let tab = choco.intArray("T",5,0,2);

let binload =
[choco.int(0,Server[0]),choco.int(0,Server[1]),choc
o.int(0,1000)];

choco.binPacking(tab,VmCapa,binload,0).post();

for it in 0..5 {
  (VmState[it] == 0) <-> (tab[it]==2);
}

for it in 0..tab.len {
 power = power + ServerPower[tab[it]];
}

power.minimize();

while(choco.solve())
 println("result: ",result,VmState,"power:
",power,tab);
```

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

QUESTION 6

1W Consumed cost 1 dollar.

We want exactly ***nbOn*** VM On and **maximize our profit**, (VMcost-Wcost)

QUESTION 6

1W Consumed cost 1 dollar.

We want exactly ***nbOn*** VM On and **maximize our profit**, (VMcost-Wcost)

(result-power).maximize();

QUESTION 7

Server power consumption = 0 if the server is unused

else, Server power consumption = N*UnitofPower

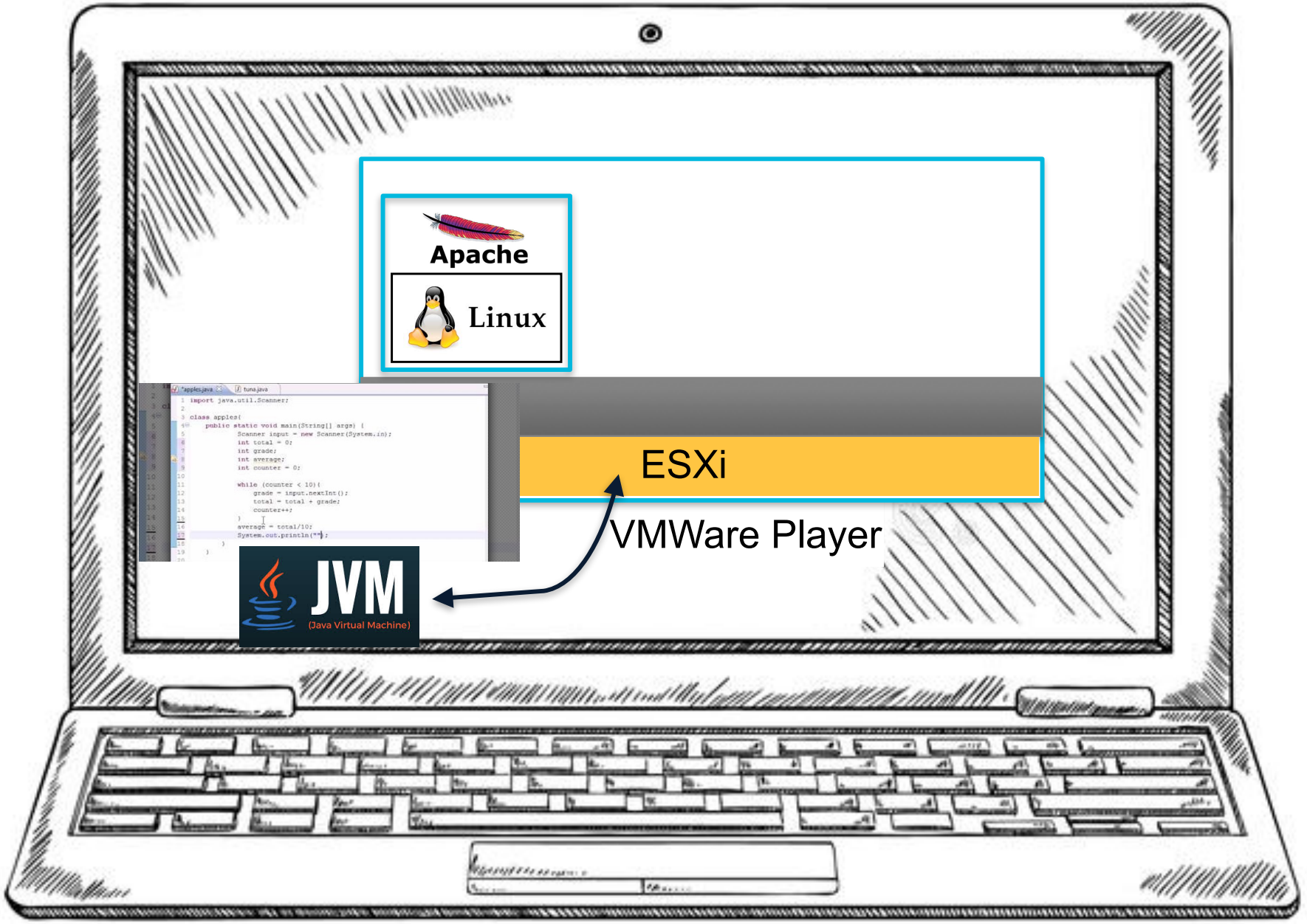
Where UnitofPower = xW for 1 unit of VM consumption

And N the total VM consumption hosted on this server

We want exactly ***nbOn*** VM On and **maximize our profit**, (VMcost-Wcost)

PRACTICAL SESSION 5

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom


Apache
Linux
import java.util.Scanner;
class apples{
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
int total = 0;
int grade;
int average;
int counter = 0;
while (counter < 10){
grade = input.nextInt();
total = total + grade;
counter++;
}
average = total/10;
System.out.println("");
}
}
JVM
(Java Virtual Machine)
ESXi
VMWare Player

http://benohead.com/getting-info-from-your-esx-server-using-the-vmware-infrastructure-vsphere-java-api-part-1/

- Do this tutorial and adapt the code to manage an ESXi Alone (without VCenter)
  - Part I : info from your ESX
  - Part II : List all VM
  - Part III : List Counters
  - Part IV: Create a VM snapshot

http://grepcode.com/file/repo1.maven.org/maven2/com.vmware/vijava/5.1/com/vmware/vim25/mo/HostSystem.java

```
> java ListHosts
Host: 'appel1806.localdomain'
Model: VMware Virtual Platform
Memory in Bytes: 4294430720
# of CPU Cores: 2
0 : Intel(R) Core(TM) i7-6567U CPU @ 3.30GHz (vendor: intel)
1 : Intel(R) Core(TM) i7-6567U CPU @ 3.30GHz (vendor: intel)
```

```
> java ListVms
Virtual Machine:tinylinux
Memory in MB: 48
# of CPUs: 1
Guest OS:Other Linux (32-bit)
VM Version:vmx-08
CPU:1 vCPU
Memory:48 MB
VMware Tools:guestToolsRunning
IP Addresses:null
State:running
Device (200): IDE 0 : IDE 0
Device (201): IDE 1 : IDE 1
Device (300): PS2 controller 0 : PS2 controller 0
Device (100): PCI controller 0 : PCI controller 0
Device (400): SIO controller 0 : SIO controller 0
Device (600): Keyboard  : Keyboard
Device (700): Pointing device : Pointing device; Device
Device (500): Video card  : Video card
Device (12000): VMCI device : Device on the virtual machine PCI bus that provides support for the virtual
machine communication interface
Device (3002): CD/DVD drive 1 : Remote ATAPI CD/DVD drive 0
Device (3000): Hard disk 1 : 65,536 KB
Device (4000): Network adapter 1 : VM Network
```

Start, Stop, Suspend a VM

CHAPTER 5
SAS
SOFTWARE AS A SERVICE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

SOFTWARE AS A SERVICE (SAAS)

- SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet.
- Saas alleviates the burden of software maintenance/ support
  - but users relinquish control over software versions and requirements.
- Terms that are used in this sphere include
  - **Platform as a Servic**e (PaaS) and
  - **Infrastructure as a Service** (IaaS)

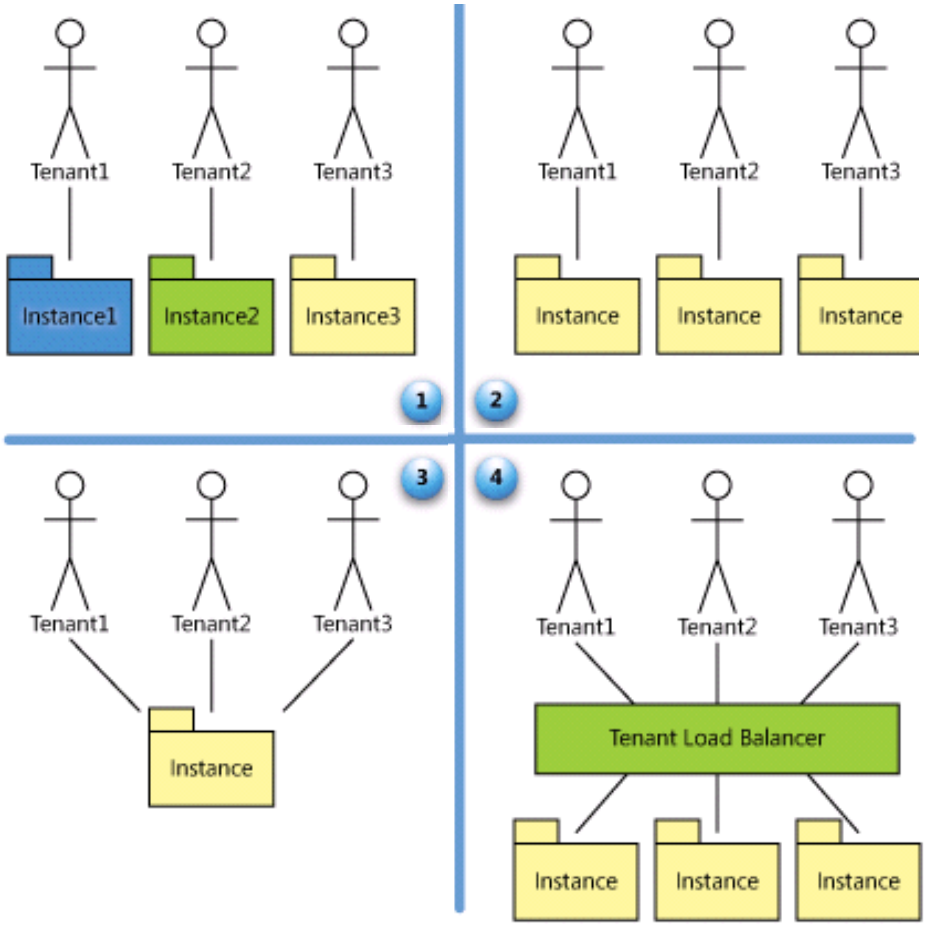
SAAS MATURITY MODEL

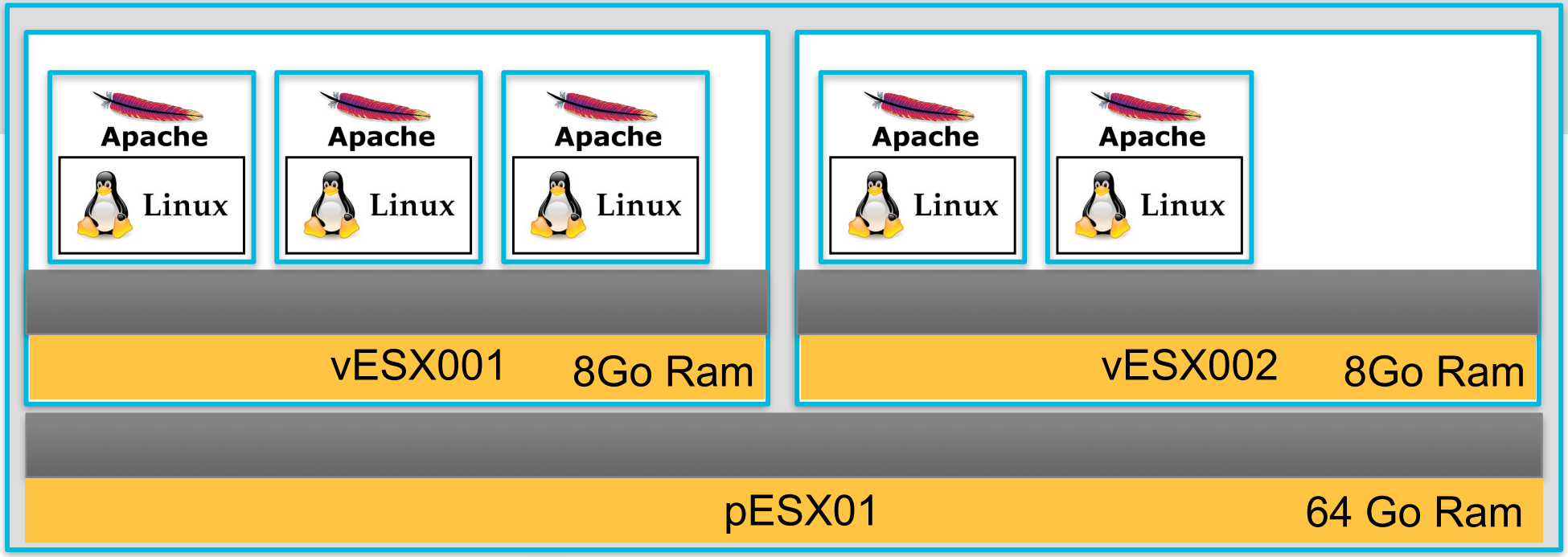
Level 1: Ad-Hoc/Custom – One Instance per customer

Level 2: Configurable per customer

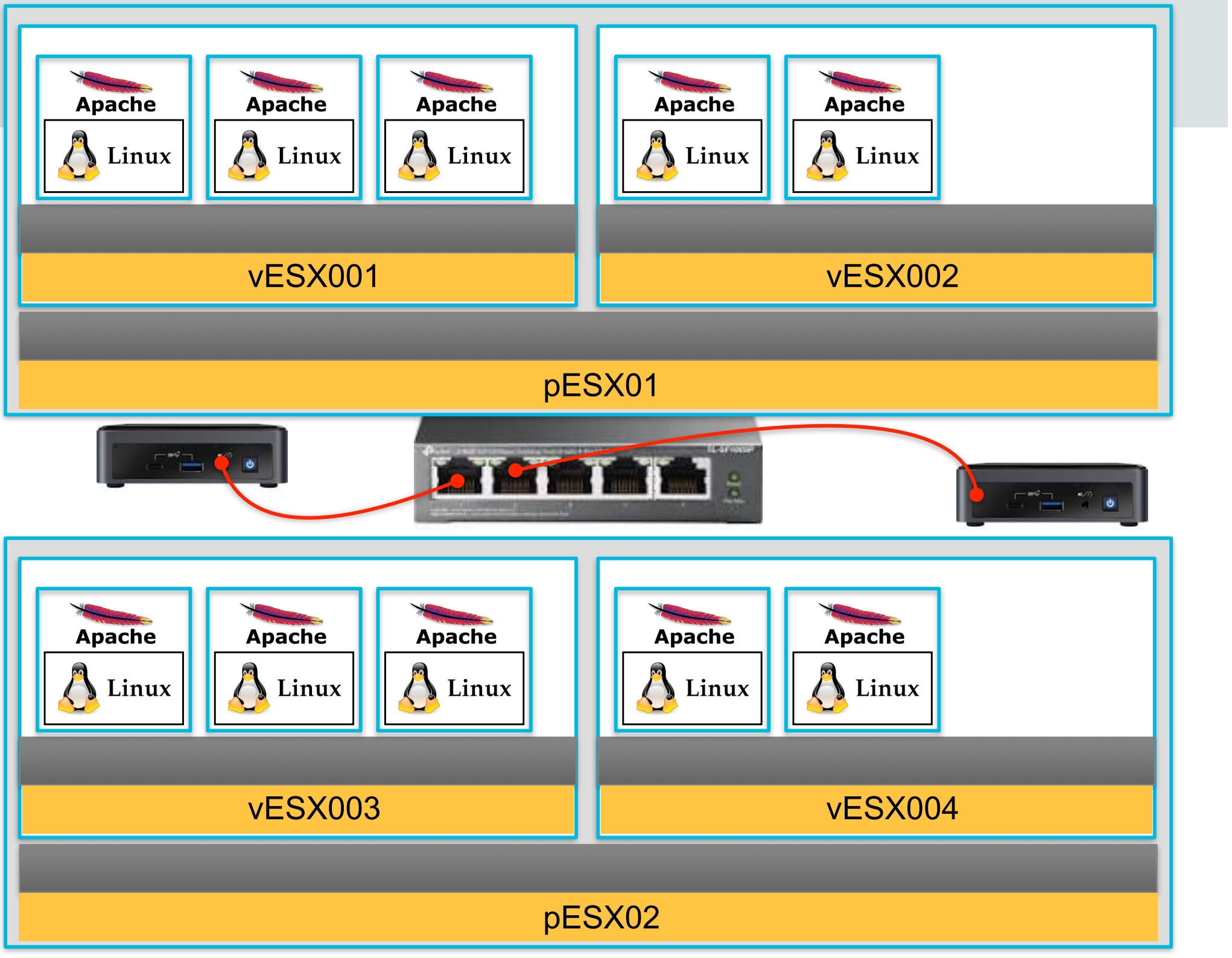
Level 3: configurable & Multi-Tenant-Efficient

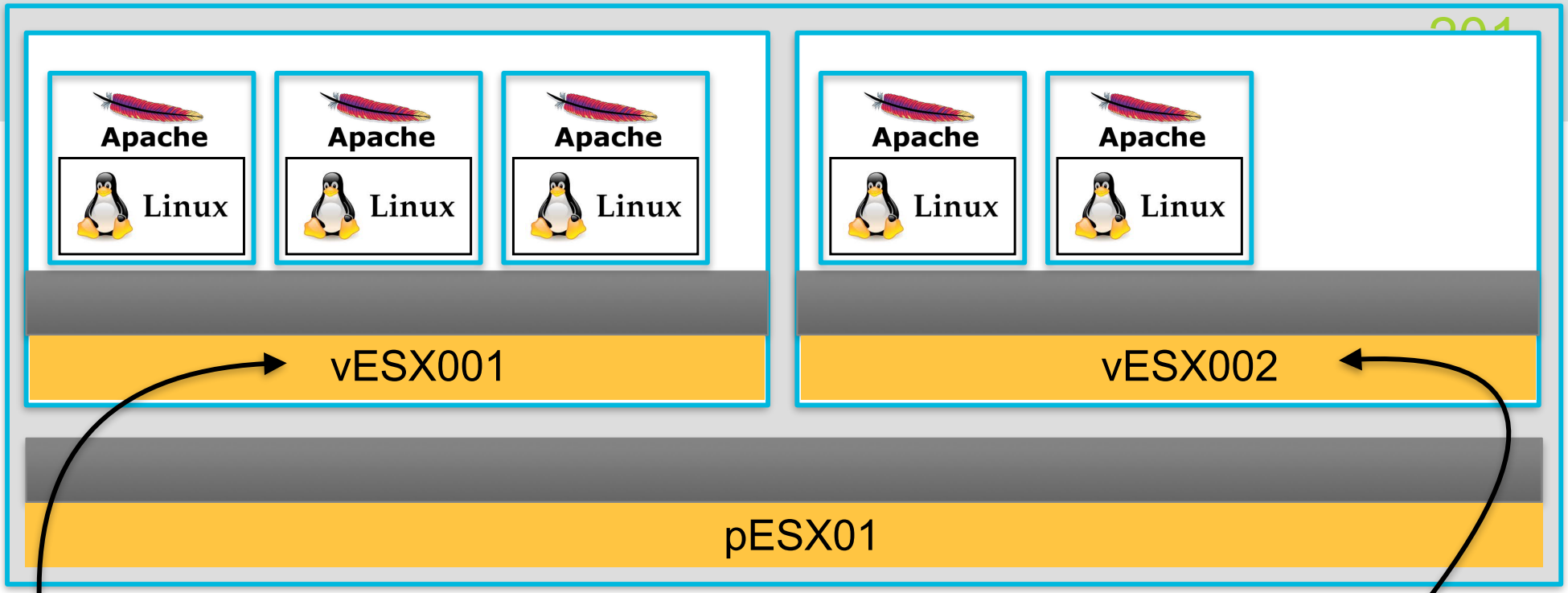
Level 4: Scalable, Configurable & Multi-Tenant-Efficient



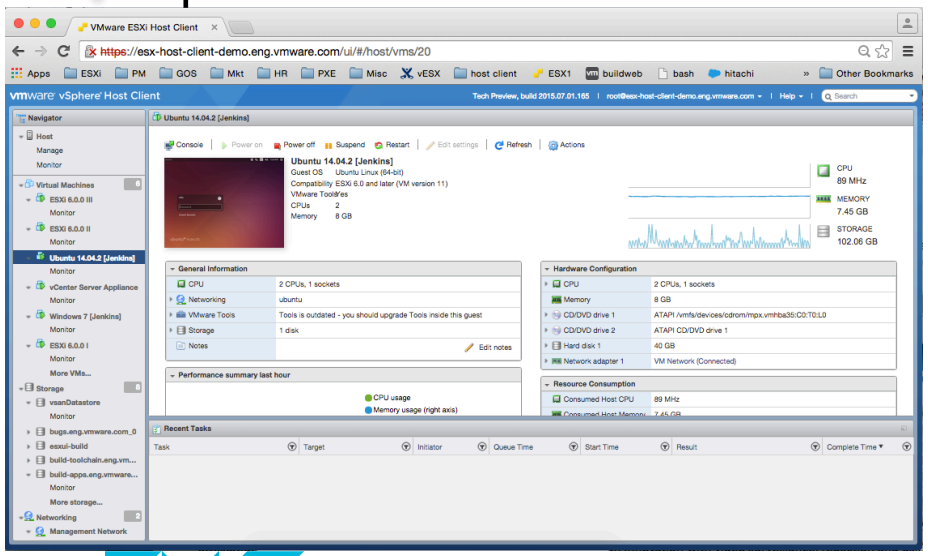


Intel i7, 64 Go Ram

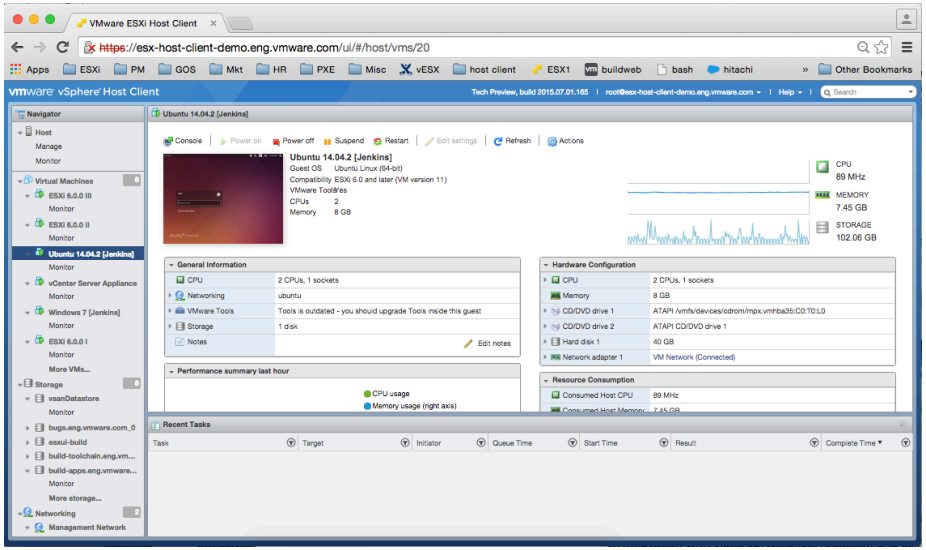

Apache
Linux
Apache
Linux
Apache
Linux
Apache
Linux
Apache
Linux
vESX001
vESX002
pESX01
Apache
Linux
Apache
Linux
Apache
Linux
Apache
Linux
Apache
Linux
vESX003
vESX004
pESX02

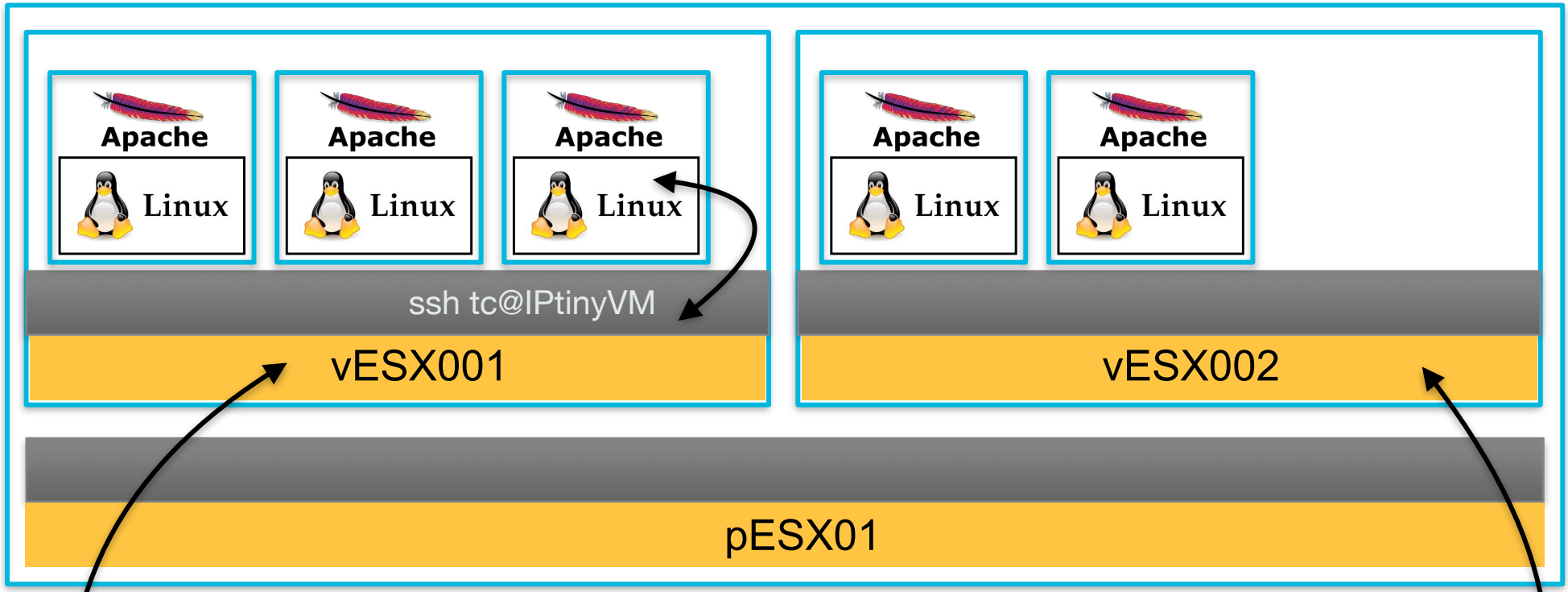


https://vesx001.cumulus.ovh/



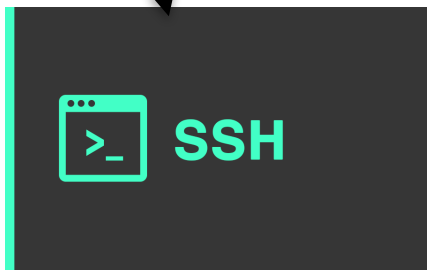
https://vesx002.cumulus.ovh/

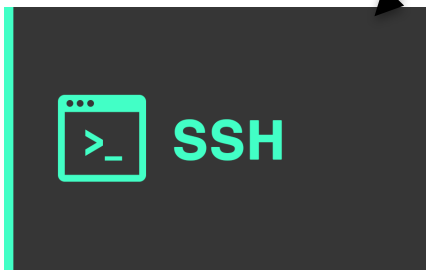


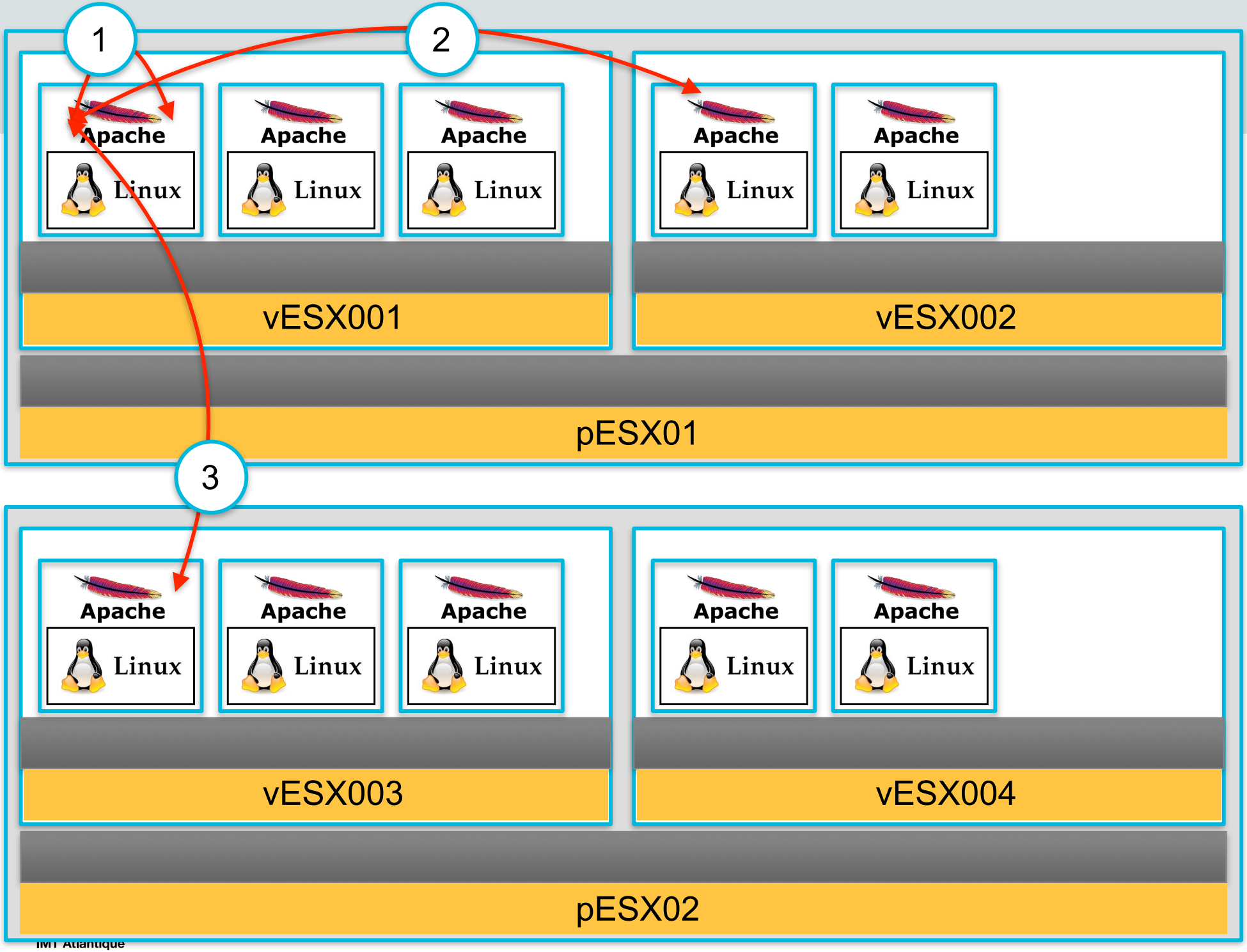



```
ssh -p 21001 root@vesx001.cumulus.ovh
```

```
ssh -p 21002 root@vesx002.cumulus.ovh
```






1
2
3
Apache
Linux
vESX001
vESX002
pESX01
vESX003
vESX004
pESX02