

---

# Cloud et calcul distribué - une introduction -

---

Sorina POP

University of Lyon, CNRS, INSERM, CREATIS, France.

**CREATIS**



# Plan du cours

---

- Introduction
- Calcul distribué sur cluster et grille
  - Clusters, gestionnaires de tâches et workflows
  - Infrastructure EGI et plateforme VIP
- Le Cloud
  - Définitions et services
  - Technologies : OpenStack, Hadoop, MapReduce, Spark
  - Cloud FranceGrilles et autres offres de Cloud
- Conteneurs (Docker)
- Big Data : introduction
- EcoInfo : quelques mots
- Conclusions

# Introduction

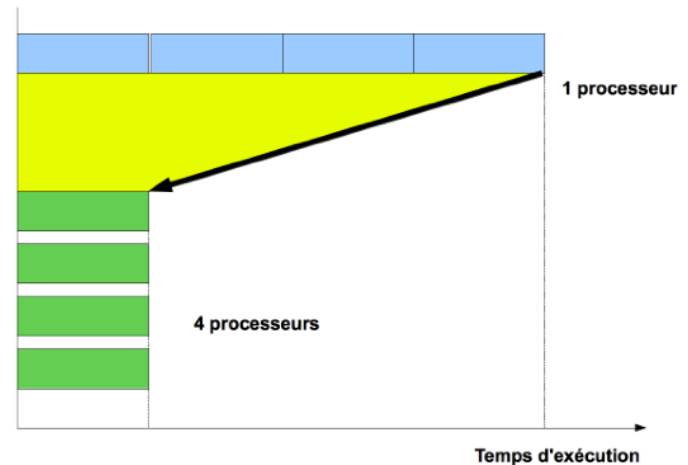
---

## ■ Pourquoi paralléliser/distribuer le calcul ?

- Algorithmes gourmands (Monte-Carlo, ...)
- Grands volumes de données, beaucoup de paramètres à explorer

## ■ Sur quelle architecture?

- Votre machine multi-coeurs
- Un supercalculateur
- Carte graphique (GPU)
- Un cluster
- Une grille ou un Cloud

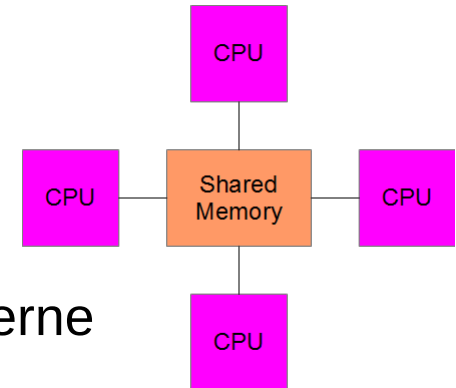


# Mémoire partagée ou distribuée ?

---

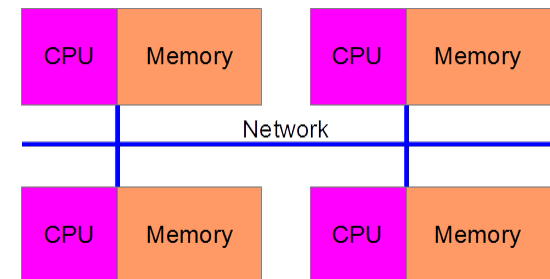
## ■ Mémoire partagée

- Tous les processeurs ont accès à un même espace mémoire
- L'accès se fait habituellement par un bus interne



## ■ Mémoire distribuée

- Chaque machine représente un nœud dans le système et a son propre espace mémoire
- L'accès à la mémoire d'une autre machine se fait par le réseau



# Calcul intensif et distribué

---

- HPC : High Performance Computing (calcul intensif)
  - Exécution d'une seule instance d'un **code parallèle** sur plusieurs processeurs
  - Tâches parallèles interconnectées (mémoire partagée ou communications)
  - Moyens de calcul adaptés : supercalculateurs, GPU, clusters, cloud
  
- HTC : High Throughput Computing (calcul distribué)
  - Exécution de plusieurs **instances indépendantes** du code (jobs, tasks) sur plusieurs processeurs potentiellement distribués géographiquement
  - Tâches indépendantes (mémoire distribuée)
  - Moyens de calcul adaptés : clusters, cloud et grilles de calcul

# Plan du cours

---

- Introduction
- **Calcul distribué sur cluster et grille de calcul**
  - Clusters, gestionnaires de tâches et workflows
  - Grille EGI et plateforme VIP
- Le Cloud
- Conteneurs (Docker)
- Big Data : introduction
- EcoInfo : quelques mots
- Conclusions

# Un cluster (grappe) de calcul

---

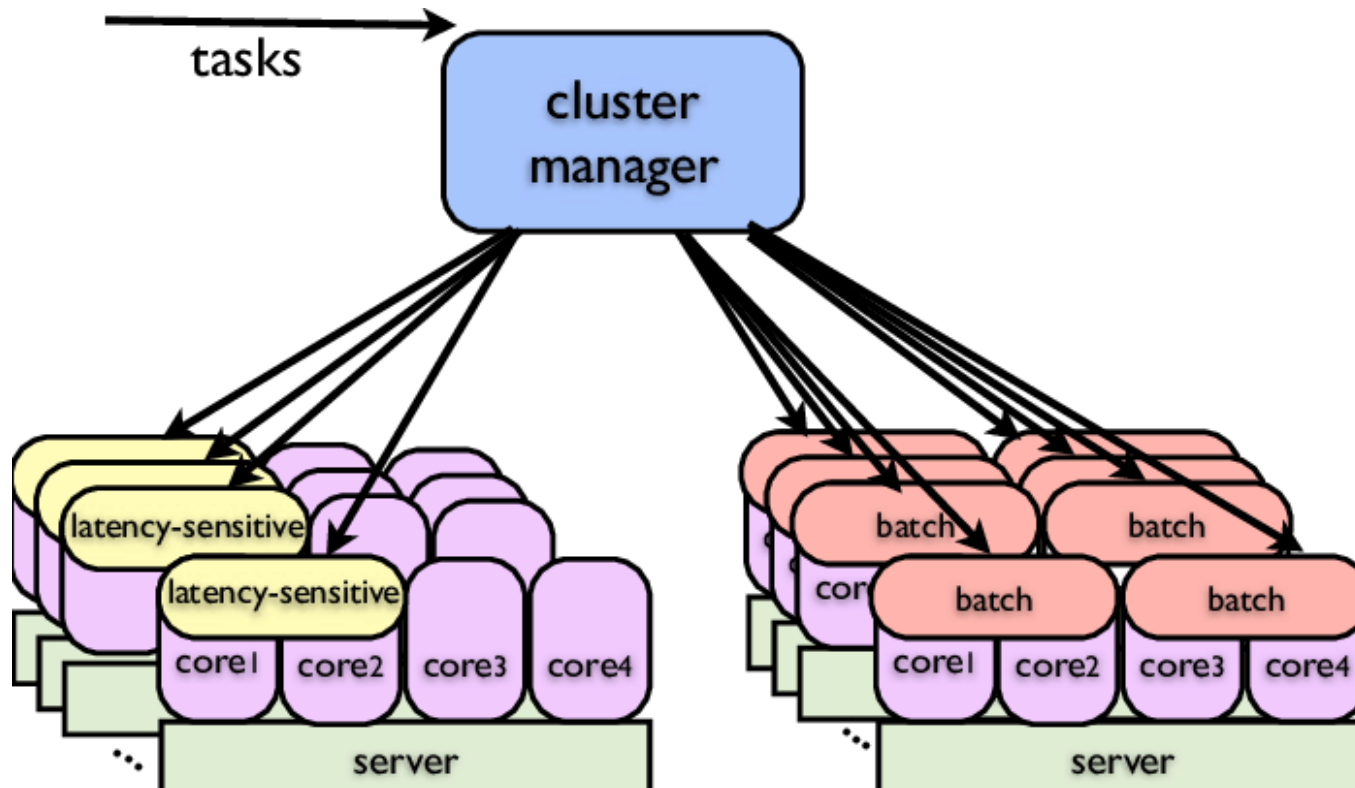
- Groupe de serveurs et d'autres ressources indépendantes fonctionnant comme un seul système
- Généralement situés à proximité et interconnectés par un réseau dédié.
- Procédé peu coûteux
- Ordinateur unique doté de plus de puissance
- Répartition des traitements sur les différents nœuds



Crédits : wikipedia

# Le gestionnaire de tâches

- Cluster manager/Batch manager/Job scheduler
  - Grid Engine (GE), Torque, SLURM
  - Organise l'exécution des tâches en respectant des règles précises



# Workflow

## ■ « Flot de travail »

- Description/décomposition d'une application en plusieurs tâches
- Facilite la parallélisation et la distribution des tâches

## ■ Langages et/ou moteurs de workflow

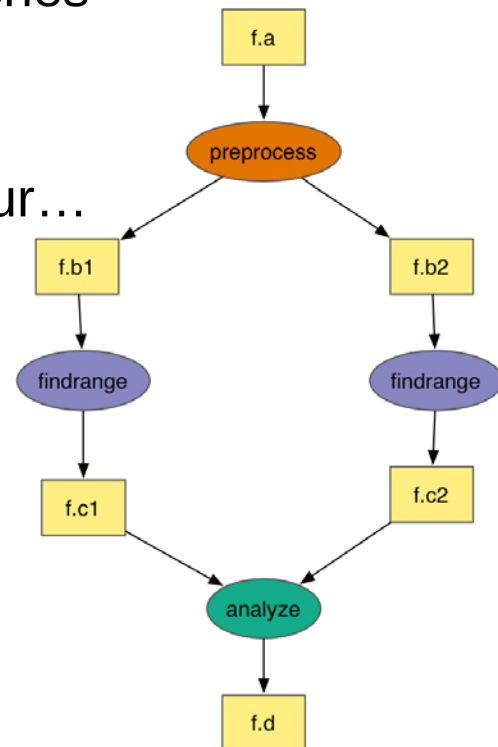
- Taverna, Nextflow, Snakemake, Pegasus, Moteur...

## ■ DAG (directed acyclic graph)

- Graphe orienté acyclique (sans circuit)

## ■ Ordonnancement (scheduling) des tâches

- Par le moteur de workflow
- Par le gestionnaire des tâches (batch manager)



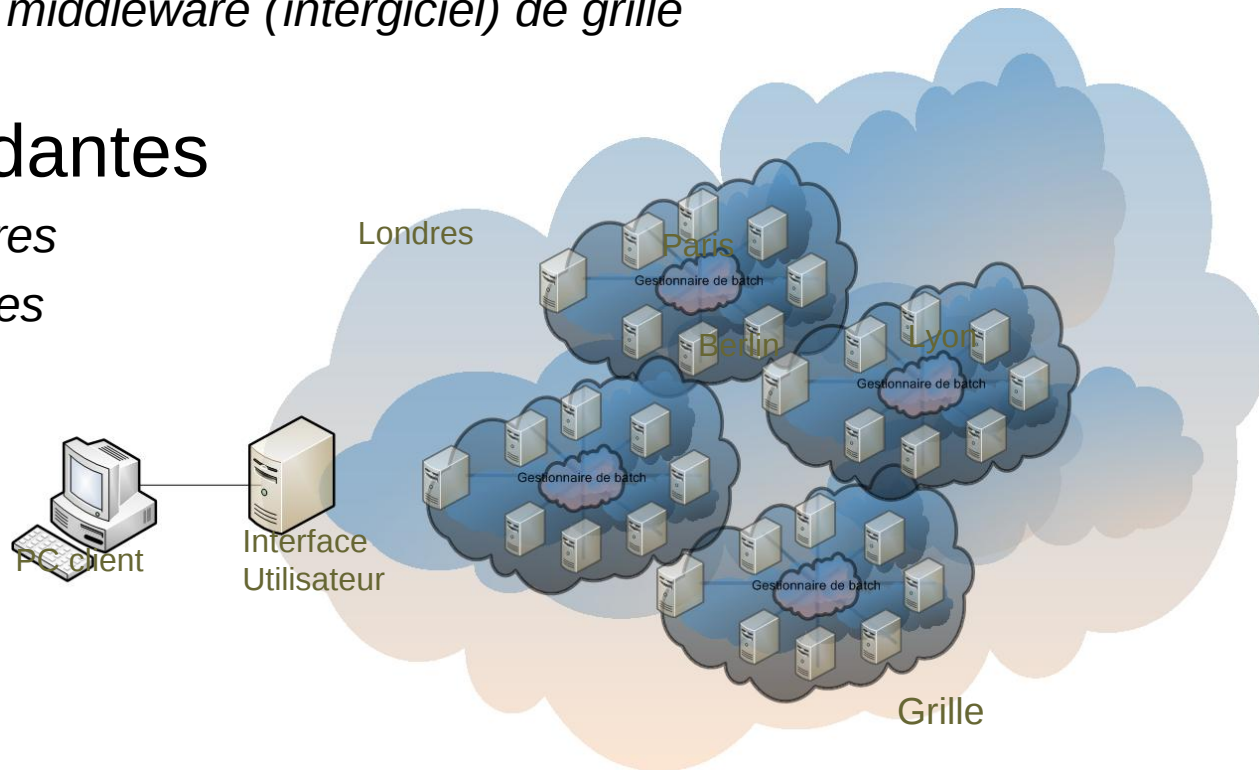
# Les grilles et le cloud

- **Caractéristiques**

- ↗ *Infrastructures informatiques destinées au calcul distribué et au stockage des données*
- ↗ *Grand nombre de machines hétérogènes et délocalisées*
- ↗ *Homogènes grâce au middleware (intergiciel) de grille*

- **Tâches indépendantes**

- ↗ *Balayage de paramètres*
- ↗ *Découpage des entrées et fusion des résultats*
- ↗ *Calculs Monte-Carlo*



# Infrastructure: EGI's biomed VO

---

- Life Sciences sector with three main thematic groups

- Medical image analysis
- Bioinformatics
- Drug discovery



<http://lsgc.org>

- EGI's biomed VO

- Computing resources
  - High Throughput Computing (large, shared Grid Computing)
  - Cloud Compute (VM-based computing with associated storage)
  - Container Compute
- Storage

- Open access

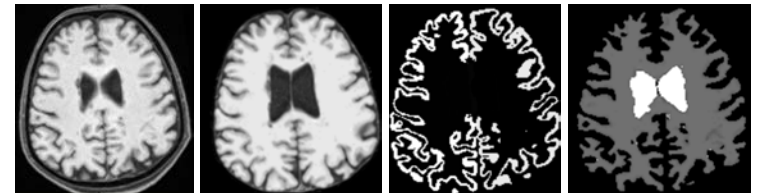
- For non-commercial users
- For life-science applications
- Acknowledgments



# Virtual Imaging Platform (VIP)

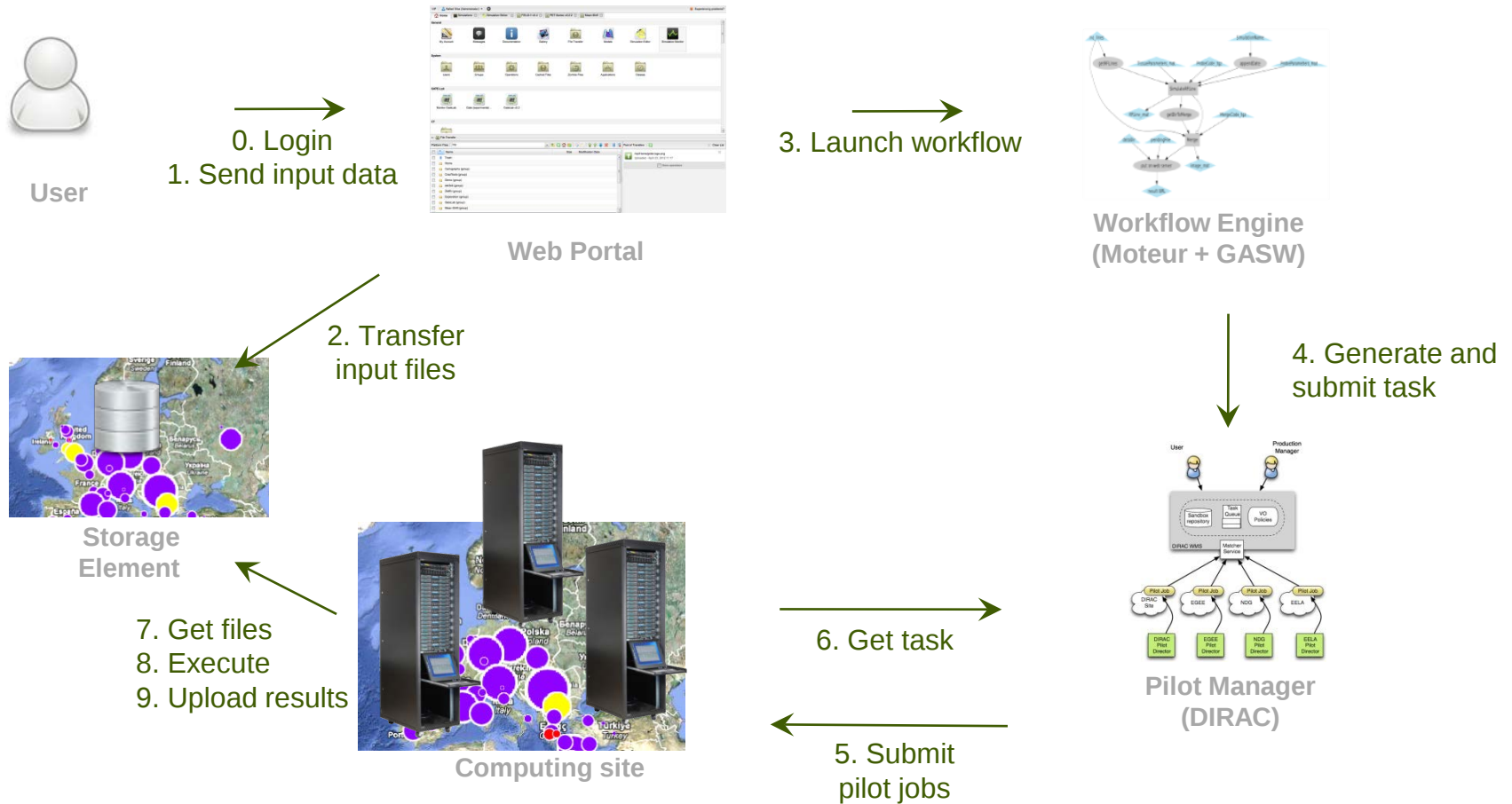
---

- Scientific applications as a Service
  - More than 20 applications publicly available
- Transparent access to computing resources
  - 395 CPU years (EGI biomed VO) used in 2019-2020
- Large community
  - More than 1450 registered users
- Open and reproducible science
  - Zenodo, DOIs, Containers, Boutiques



Example of white/grey matter brain segmentation with [Freesurfer](#) on VIP  
Credits : Berardino Barile and Dominique Sappey-Mariniere, Creatis

# VIP architecture



# Monte-Carlo simulations on VIP

## ■ Repeated random sampling of solution space

- Large number of independent events

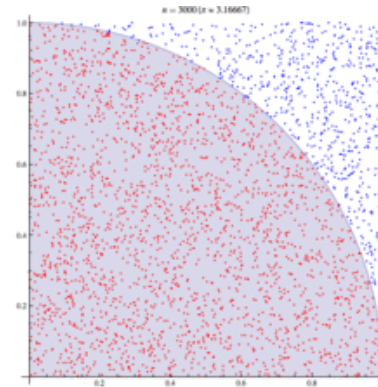
## ■ Parallelisation: split and merge

- Define a domain of possible inputs.
- Process the inputs.
- Merge the results.

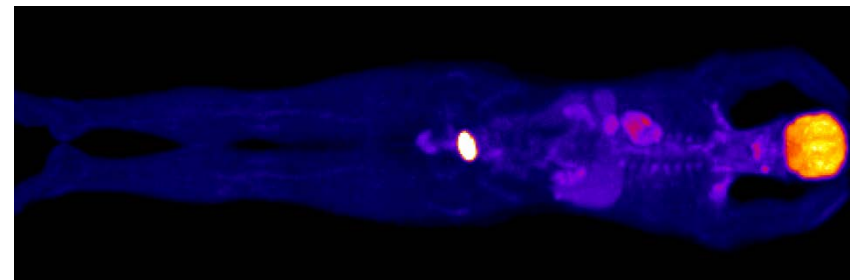
- Example: GATE simulator

- Simulation of particle tracking
- Nuclear medicine simulations for

Positron Emission Tomography (PET)  
and radiation therapy imaging



Computing  $\pi$   
Source: Wikipedia

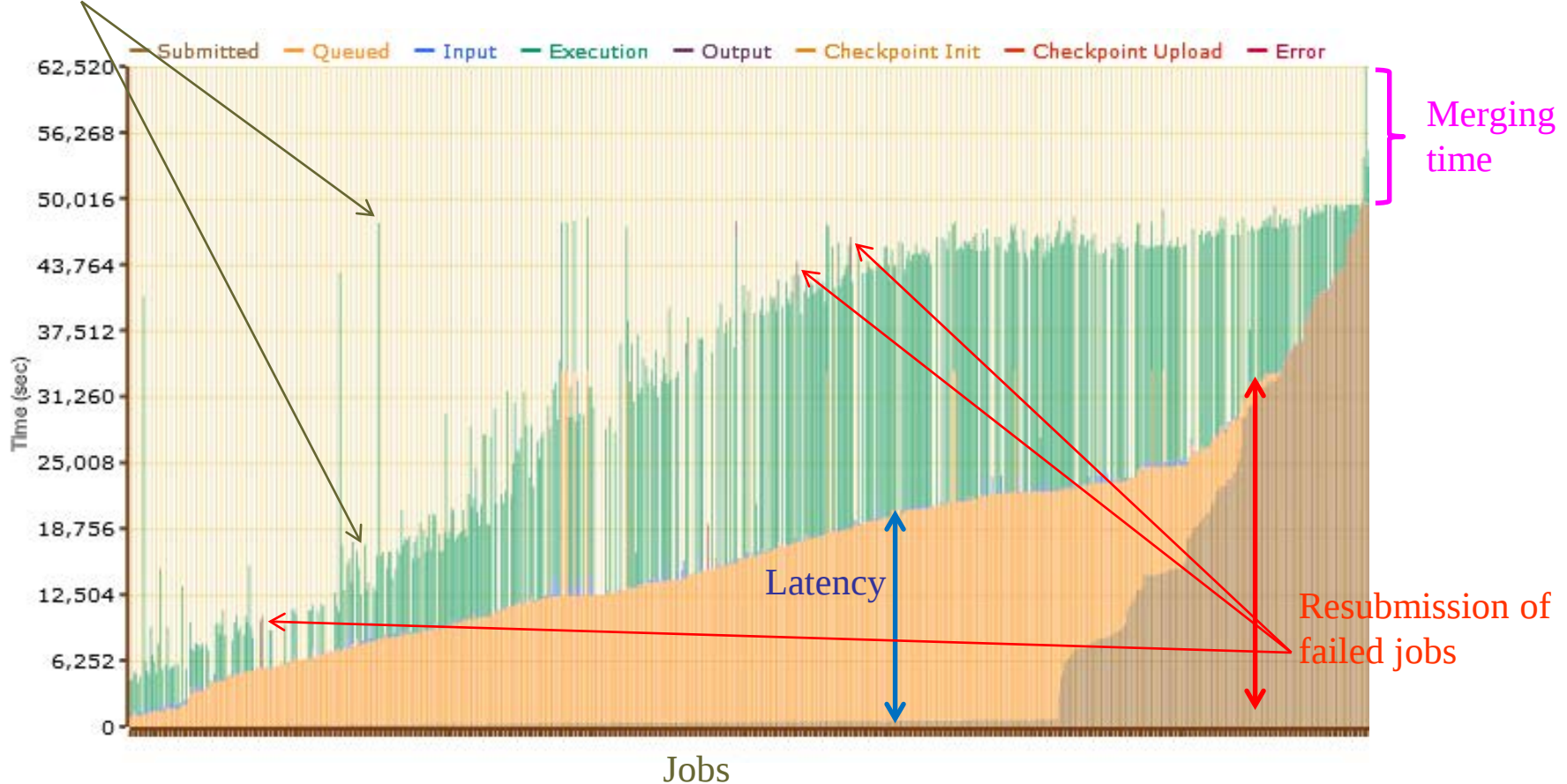


3D whole-body F18-FDG PET scan  
simulated with GATE, representing 4,000  
CPU hours (5.3 months). Credits: IMNC-  
IN2P3.

# GATE on VIP (I)

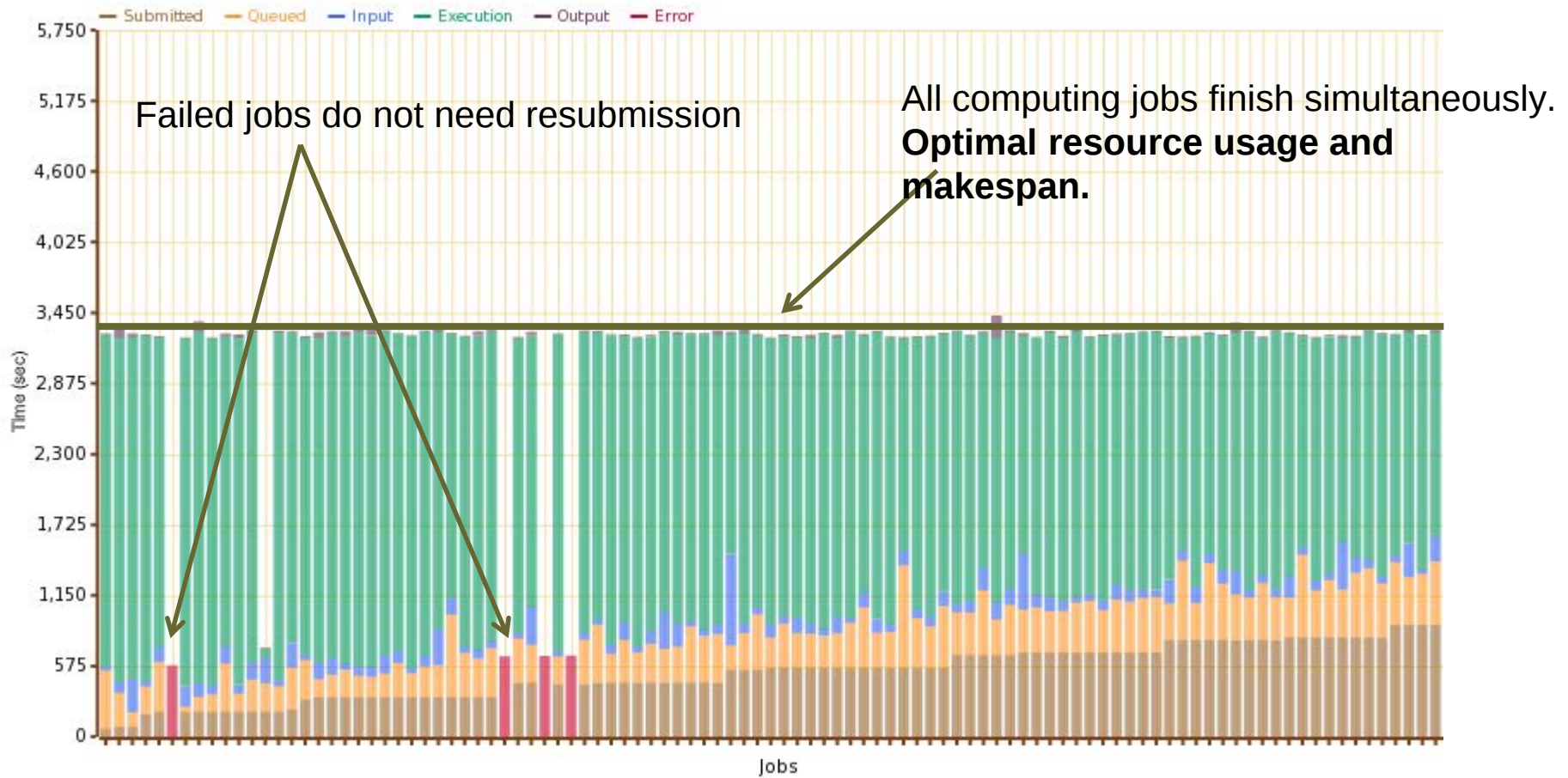
- Static scheduling -

Heterogeneity (same start, different finish time)



# GATE on VIP (II)

- Dynamic scheduling -



# Plan du cours

---

- Introduction
- Calcul distribué sur cluster et grille de calcul
- **Le Cloud**
  - Définitions et services
  - Technologies : OpenStack, Hadoop, MapReduce, Spark
  - Cloud FranceGrilles et autres offres de Cloud
- Conteneurs (Docker)
- Big Data : introduction
- EcoInfo : quelques mots
- Conclusions

# Cloud Computing

---

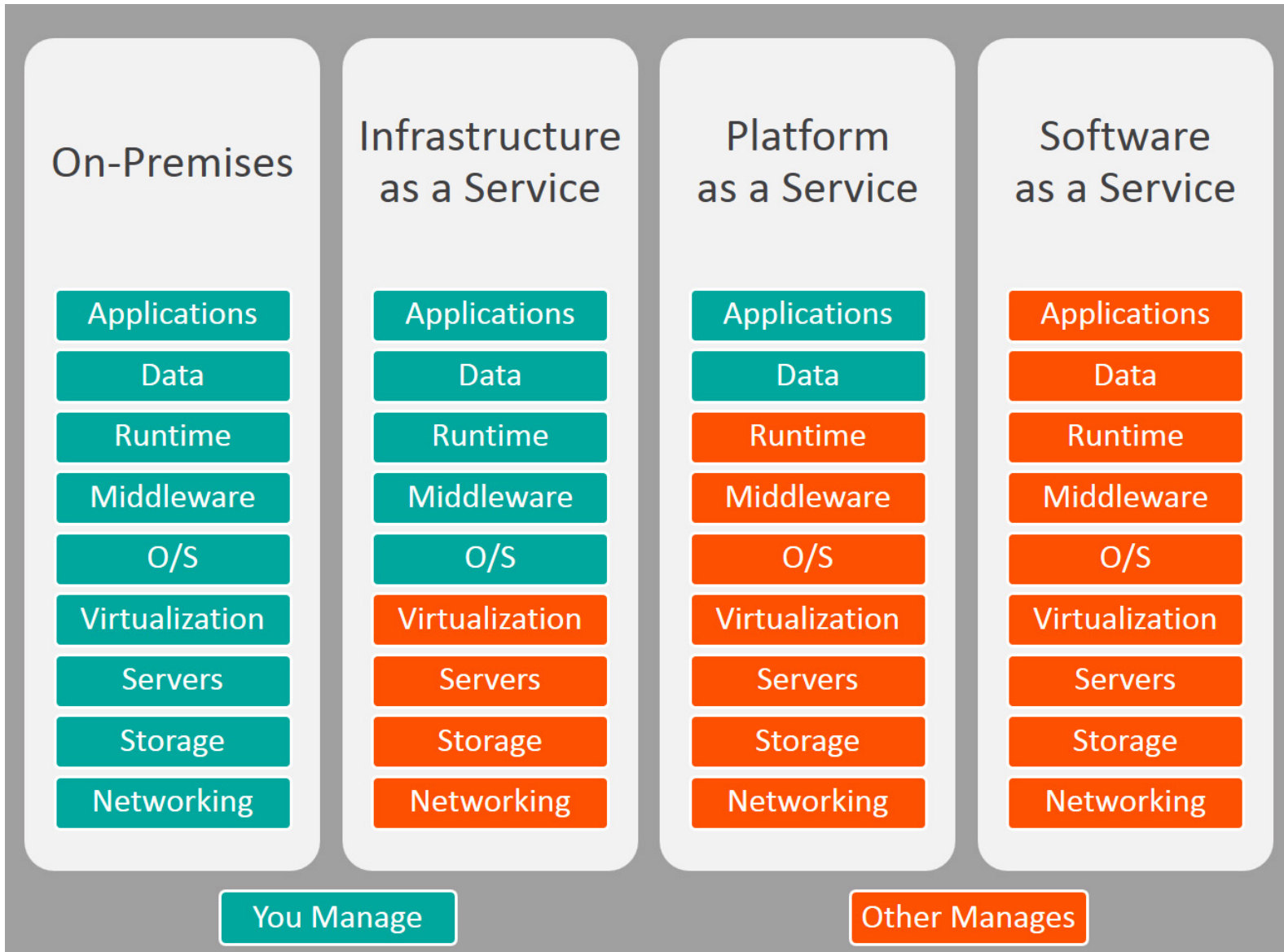
## ■ Definition

- A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

## ■ In addition to Grids, the Cloud provides

- On demand services, that are always on, anywhere and anytime
- Economic model allowing to pay as much as used and needed
- Virtualization (abstraction of a physical host machine)

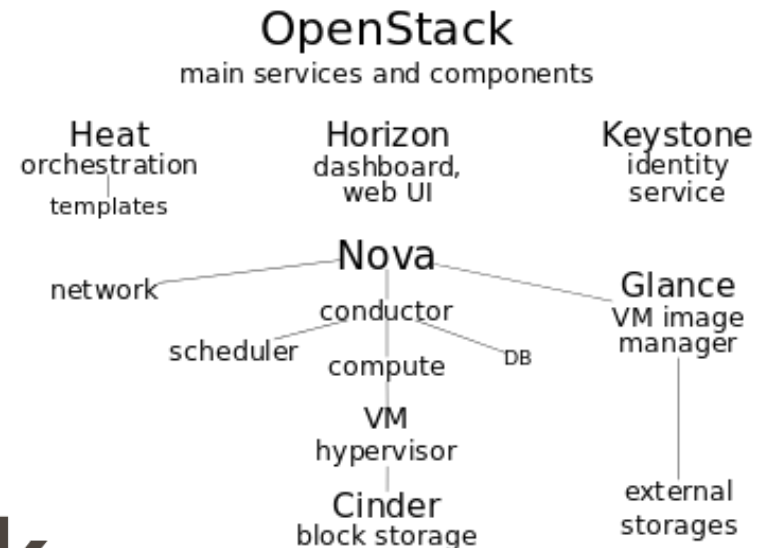
# Cloud service models



Source : <https://www.business2community.com/cloud-computing/8-ways-aws-beats-azure-in-the-cloud-02190398>

# OpenStack

- OpenStack is a free and open-source software platform for cloud computing, mostly deployed as an IaaS
- Interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center
- Users can manage it through
  - a web-based dashboard
  - command-line tools
  - a RESTful API.



# Horizon (OpenStack)

## ■ OpenStack dashboard

- Web based user interface to OpenStack services including Nova, Swift, Keystone, etc.

openstack. Default • FG\_formation ▾

Projet ▾

Accès API

Compute ▾

Vue d'ensemble

**Instances**

Images

Paires de clés

Volumes >

Réseau >

Orchestration >

Identité >

Projet / Compute / Instances

### Instances

ID de l'instance = ▾  Filtrer Lancer une instz

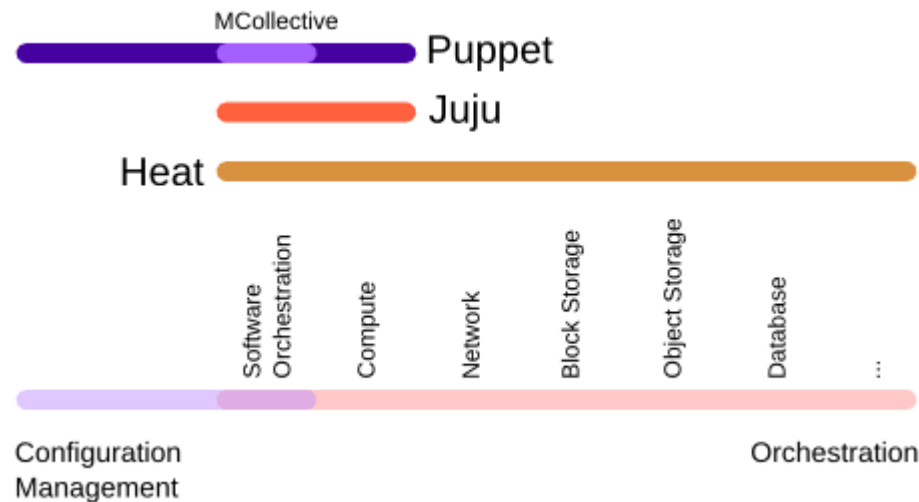
Affichage de 2 éléments

| <input type="checkbox"/> | Nom de l'instance                            | Nom de l'image               | Adresse IP                                        | Gabarit   | Paire de clés | Statut | Zone de disponibilité | Tâche | État de l'alimentation |
|--------------------------|----------------------------------------------|------------------------------|---------------------------------------------------|-----------|---------------|--------|-----------------------|-------|------------------------|
| <input type="checkbox"/> | TestHeatWithNetwork-my_instance-eduisbajbwzl | FG_formation-CentOS-7-x86_64 | 172.16.7.22<br>IP flottantes :<br>134.158.151.125 | m1.medium | tpinsa        | Active | nova                  | Aucun | En fonctionnement      |
| <input type="checkbox"/> | TestHeatSorina                               | FG_formation-CentOS-7-x86_64 | 172.16.7.6<br>IP flottantes :<br>134.158.151.135  | m1.medium | tpinsa        | Active | nova                  | Aucun | En fonctionnement      |

# Orchestration

---

- Automate the deployment and configuration of infrastructure
  - Build your own collection of resources: instances (VMs), networks, security groups, and even auto-scaling rules



<https://www.zerobanana.com/archive/2014/05/08#heat-configuration-management>

# Heat (OpenStack)

---

## ■ OpenStack Orchestration

- Launch multiple composite cloud applications based on templates in the form of text files that can be treated like code (infrastructure as code)

## ■ How it works

- You write Heat template (text file) describing the infrastructure (servers, floating ips, volumes, security groups, users, etc)
- Templates can be deployed through the Horizon Dashboard or with OpenStack CLI
- Heat builds and manages the infrastructure described in the template
- <https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/metacloud/newbie-tutorial-heat.pdf>

# Hadoop

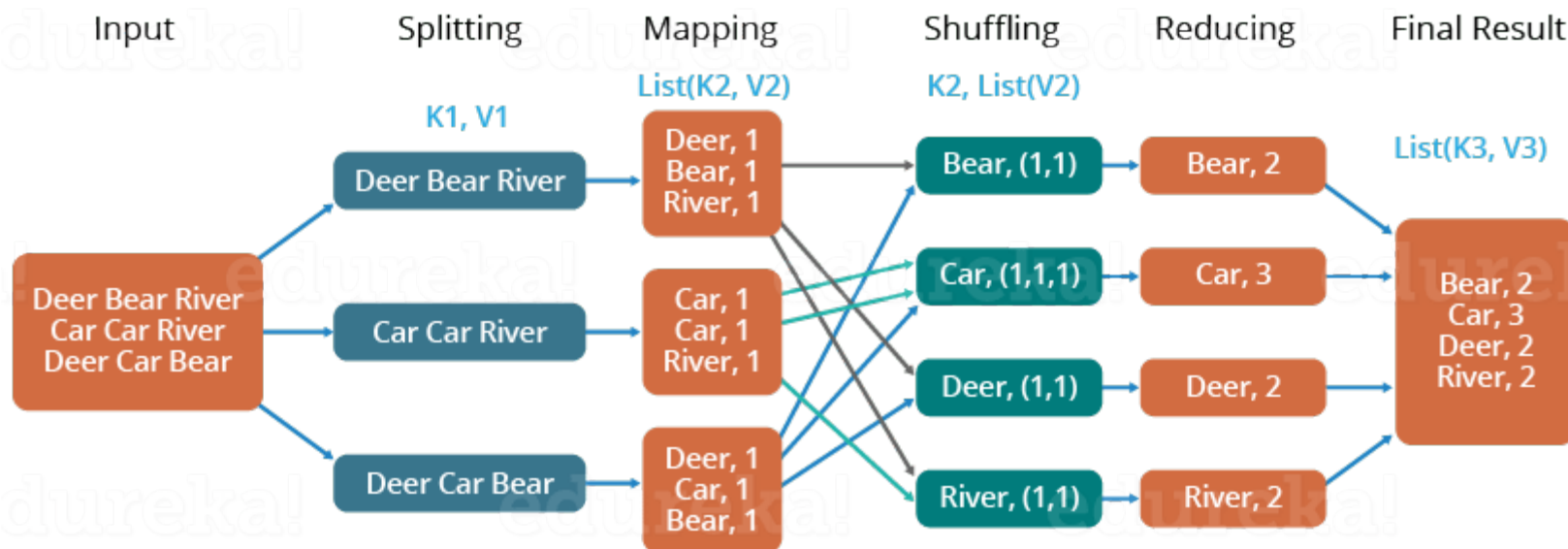
---

- Open-source software for distributed computing
  - Framework for the distributed processing of large data sets using simple programming models
  - Designed to detect and handle failures at the application layer
- **Hadoop Distributed File System (HDFS)**
  - Distributed file system for high-throughput access to data
- **Hadoop YARN**
  - Job scheduling and cluster resource management.
- **Hadoop MapReduce**
  - A YARN-based system for parallel processing of large data
  - Take advantage of data locality
  - Libraries in many prog languages

# Map Reduce Example

## The Overall MapReduce Word Count Process

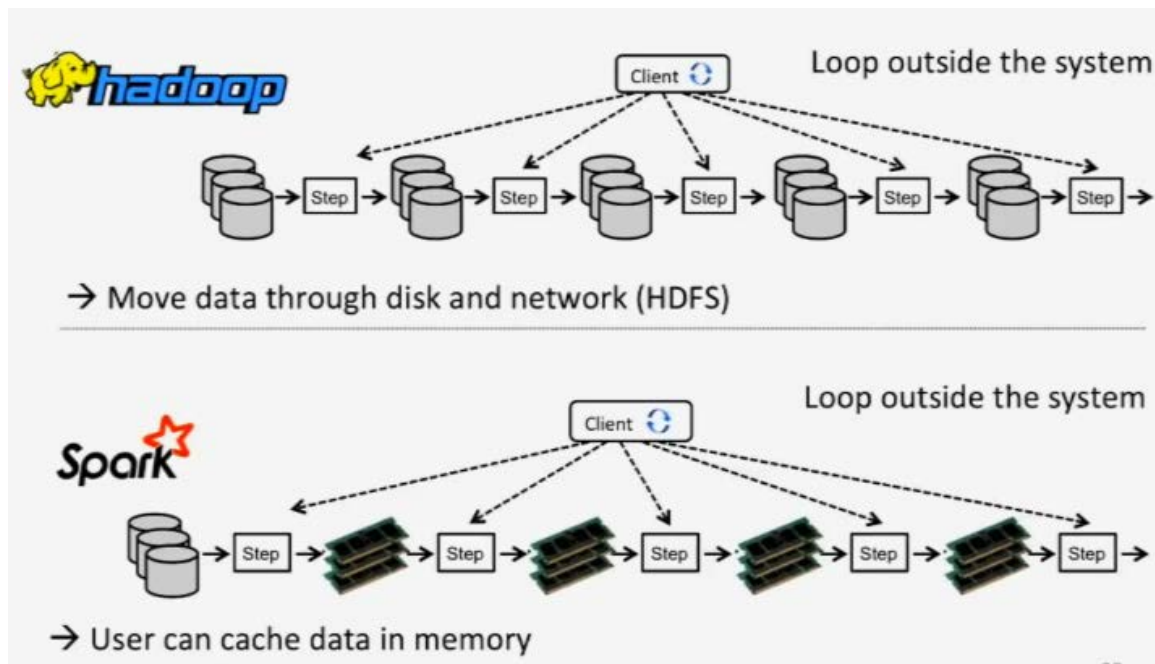
edureka!



Crédits : <https://www.edureka.co/blog/mapreduce-tutorial/>

# Spark

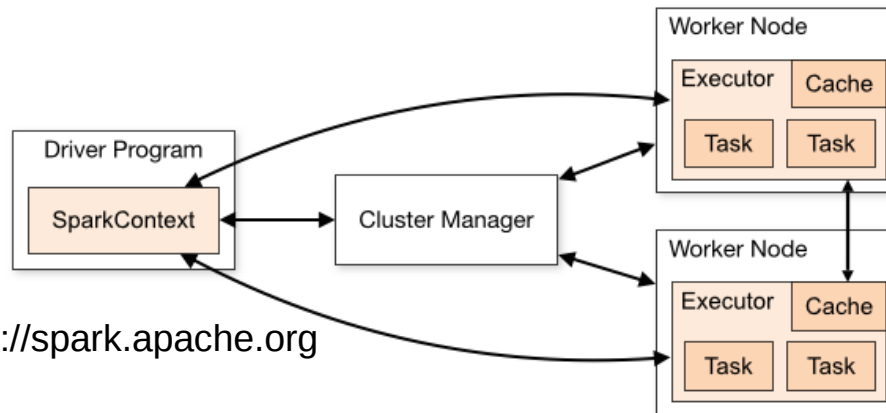
- Apache Spark - a replacement of Hadoop MapReduce ?
  - Spark supports in-memory computing, which is faster than MapReduce's file-based model
  - Spark's programming model is much richer than MapReduce
  - Spark can run on a variety of clusters, including Hadoop



Source : <https://www.lemondeinformatique.fr>

# How Spark Works

- Spark uses a master/slave architecture
  - One central **driver** and many distributed workers (**executors**)
  - The driver and each executor are separate java processes

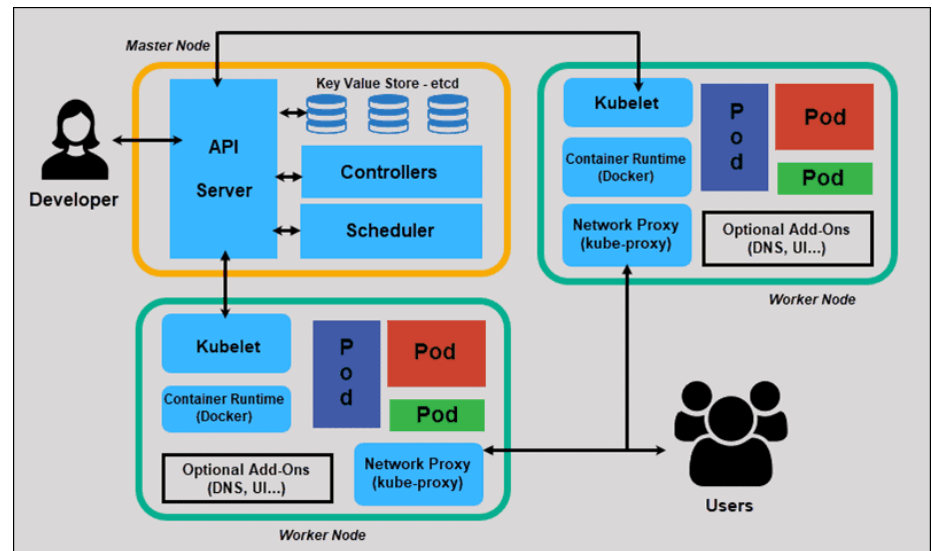


Credit: <https://spark.apache.org>

- Cluster Manager Types
  - Standalone – a simple cluster manager included with Spark that makes it easy to set up a cluster
  - Hadoop YARN – the resource manager in Hadoop
  - Others, such as Apache Mesos and Kubernetes

# Kubernetes

- Open source platform for managing containerized workloads and services
  - Note: containers will be discussed later in the presentation
- The basic scheduling unit in Kubernetes is a pod
  - One or more containers that are guaranteed to be co-located on the same node
- Worker node
  - Machine where containers (workloads) are deployed



Kubernetes architecture and components

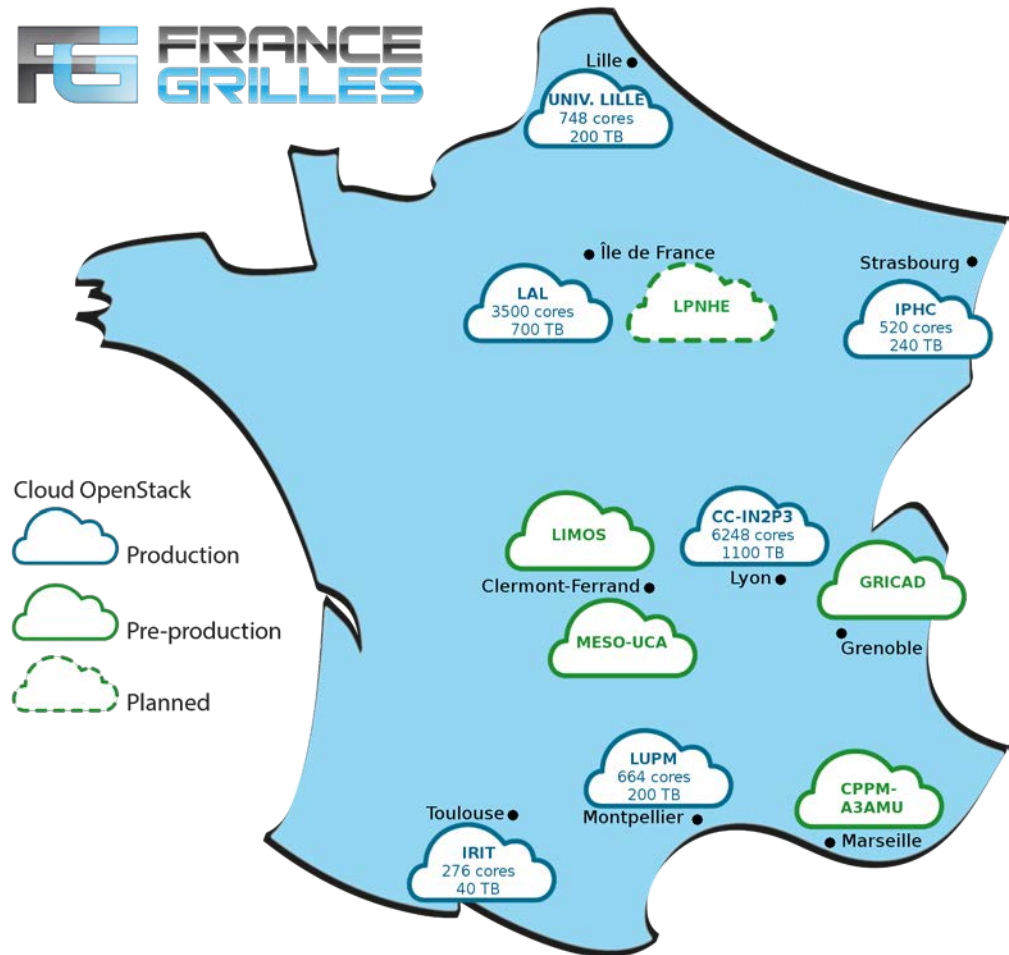
Source: <https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams>

# Cloud France-Grilles

## ■ Cloud académique pour calcul scientifique

- 3882 cœurs de CPU
- 14 TB de RAM
- 560 TB de stockage

Crédits : Jérôme Pansanel  
(Directeur technique FG)





## ■ Vision

- Provide a web of FAIR (Findable, Accessible, Interoperable, Reusable) data and services for science in Europe
- Give all European researchers access to (free) scientific data, services and e-infrastructures to analyse data
- Rely mainly on existing services and e-infrastructures

## ■ Portal: <https://eosc-portal.eu>

## ■ Community

- Bring together researchers and engineers from all scientific domains and EU countries

## ■ Partnership

- Three partners: the European Commission, the EU member states and the users & providers

# Autres offres de Cloud

---

## ■ Amazon (AWS)

- Elastic Compute Cloud (EC2)
- Drive (stockage)

## ■ Microsoft

- Azure

## ■ Google

- Google Compute Engine (GCE)
- Google Cloud Storage

■ ...

# AWS vs Azure vs Google

---

## PUBLIC CLOUD



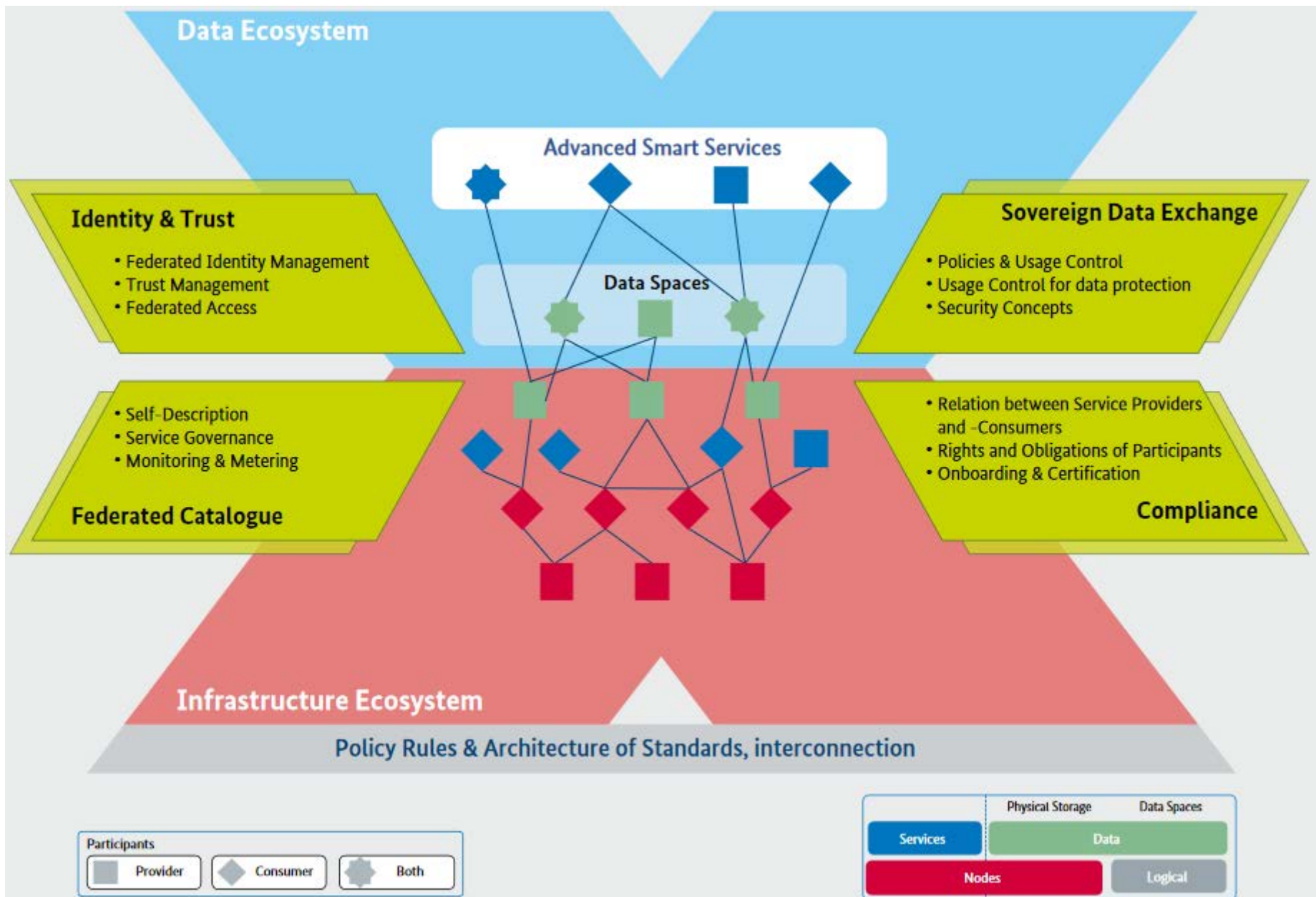
Source : <http://cloudacademy.com/blog/public-cloud-war-aws-vs-azure-vs-google/>  
<https://stackify.com/microsoft-azure-vs-amazon-web-services-vs-google-compute-comparison/>

# Gaia-X

---

- Ecosystème de cloud européen
  - Etablir des règles et des standards communs
  - Fonctionnement décentralisé et fédération des fournisseurs de services et d'infrastructures existantes
  - Favoriser la portabilité et l'interopérabilité
- Initié en Allemagne fin 2019, rejoint par la France
- Premiers cas d'usage
  - fixer les règles (les attributs retenus, les standards techniques associés, etc.)
  - trouver comment contrôler la bonne application de ces règles
  - définir le standard commun qui sera utilisé pour favoriser l'interopérabilité.

Source : <https://www.usine-digitale.fr/editorial/qu-est-ce-que-gaia-x-le-projet-franco-allemand-de-cloud-europeen.N971806>



# EOSC vs GAIA-X

---

## ■ EOSC

- Web of FAIR Digital Objects and Related Services for Science
- FAIR principles and metadata standards as guidelines for interoperability and maximum sharing

## ■ GAIA-X

- next generation of a EU data infrastructure
  - digital sovereignty while promoting innovation
  - strong focus on enabling data sharing in business sectors
- ## ■ Compatible approaches with many different potential touching points

# Plan du cours

---

- Introduction
- Calcul distribué sur grille
- Le Cloud
- **Conteneurs (Docker)**
- Big Data : introduction
- EcoInfo : quelques mots
- Conclusions

# Pourquoi s'intéresser aux conteneurs

---

## ■ Portabilité et compatibilité

- Ex : j'ai développé un super algo de segmentation
  - Comment le faire tourner sur différents clusters ?
  - Comment le partager avec mes collaborateurs ?

## ■ Reproductibilité

- Ex : j'ai obtenu des super résultats que je veux publier
  - Comment assurer leur reproductibilité dans 1 an ?
  - Comment permettre leur reproduction par les reviewers/chercheurs intéressés ?
    - Sans forcément fournir le code

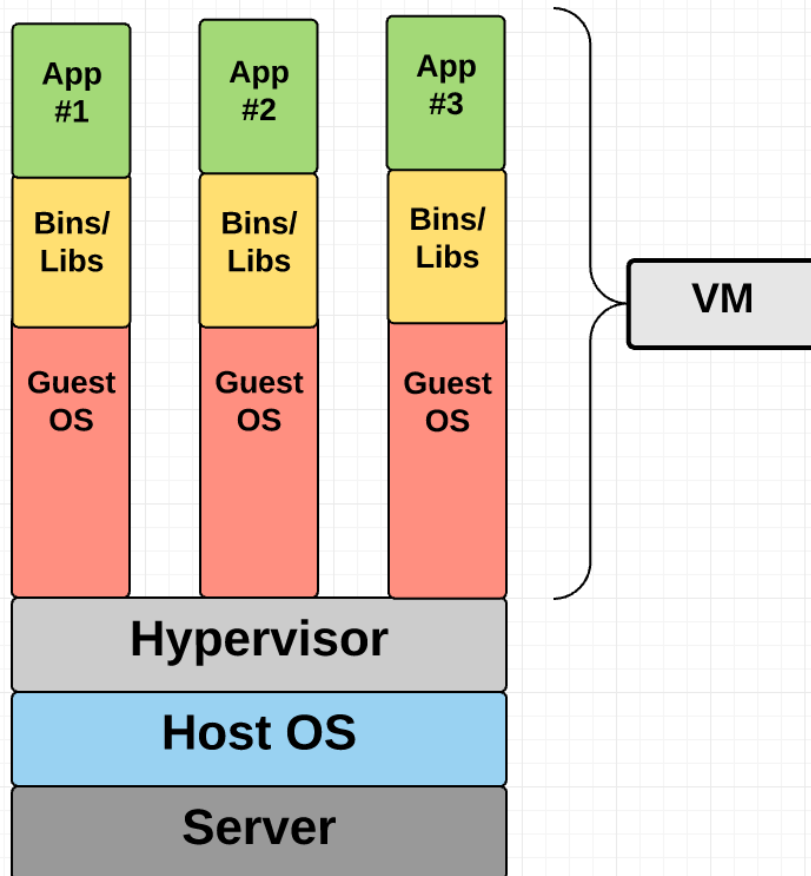
# But what are containers?...

---

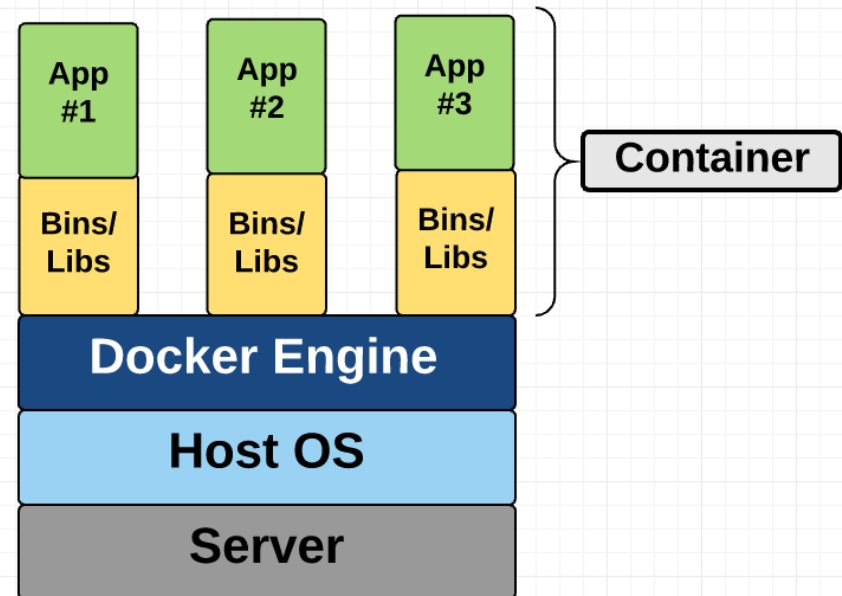
- A container = an entire runtime environment
  - An application + all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package
  - By containerizing the application platform and its dependencies, differences in OS distributions and underlying infrastructure are abstracted away
- Docker has become synonymous with container technology because it has been the most successful at popularizing it, but
  - Container technology is not new
  - Other containers exist (Singularity, Charliecloud ...)

# Virtualisation et conteneurs

- Une VM (Virtual Machine) « imite » intégralement un serveur et contient un système d'exploitation complet



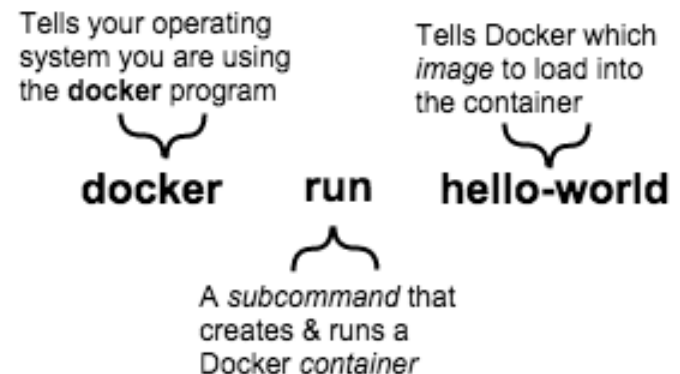
- Le concept de conteneurisation permet aux instances virtuelles de partager un système d'exploitation hôte unique. Le rôle de l'hyperviseur est assuré par un moteur de conteneurisation (par ex, Docker)



# Docker images and containers

---

- A Docker **image** is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.
  - A docker image is described in text file called a *Dockerfile*
  - You can also create an image from a container (docker commit)
- A Docker **container** is a runtime instance of an image
  - What the image becomes in memory when executed
  - You can see a list of your running containers with the command, `docker ps`, just as you would in Linux
  - You can run, start, stop, move, or delete a container using Docker API or CLI commands.



Source: <https://docs.docker.com>

# Dockerfile

---

- *# Use an official centos as a parent image*
- from centos:latest (attention, “latest” not a good idea for reproducibility)
- *# Install any needed packages*
- RUN yum install -y epel-release git gcc make
- RUN git clone git://git.creatis.insa-lyon.fr/demoSorina
- *# Set the working directory to / demoSorina*
- WORKDIR /demoSorina
- RUN make

# Demo

---

■ *# Build the image (once)*

■ \$ docker build . -t camarasu/demo\_docker:0.1

■ *# Run image with mounted volumes (multiple times)*

■ \$ docker run --rm -v \$PWD:/dirOut -it  
camarasu/demo\_docker:0.1 /bin/bash

■ [root@c30117fdeb4e demoSorina]# ls

■ Makefile hello hello.c hello.o

■ [root@c30117fdeb4e demoSorina]# ls /dirOut/

■ Dockerfile notes.txt

# Plan du cours

---

- Introduction
- Calcul distribué sur cluster et grille
- Le Cloud
- Conteneurs (Docker)
- **Big Data : introduction**
- EcoInfo : quelques mots
- Conclusions

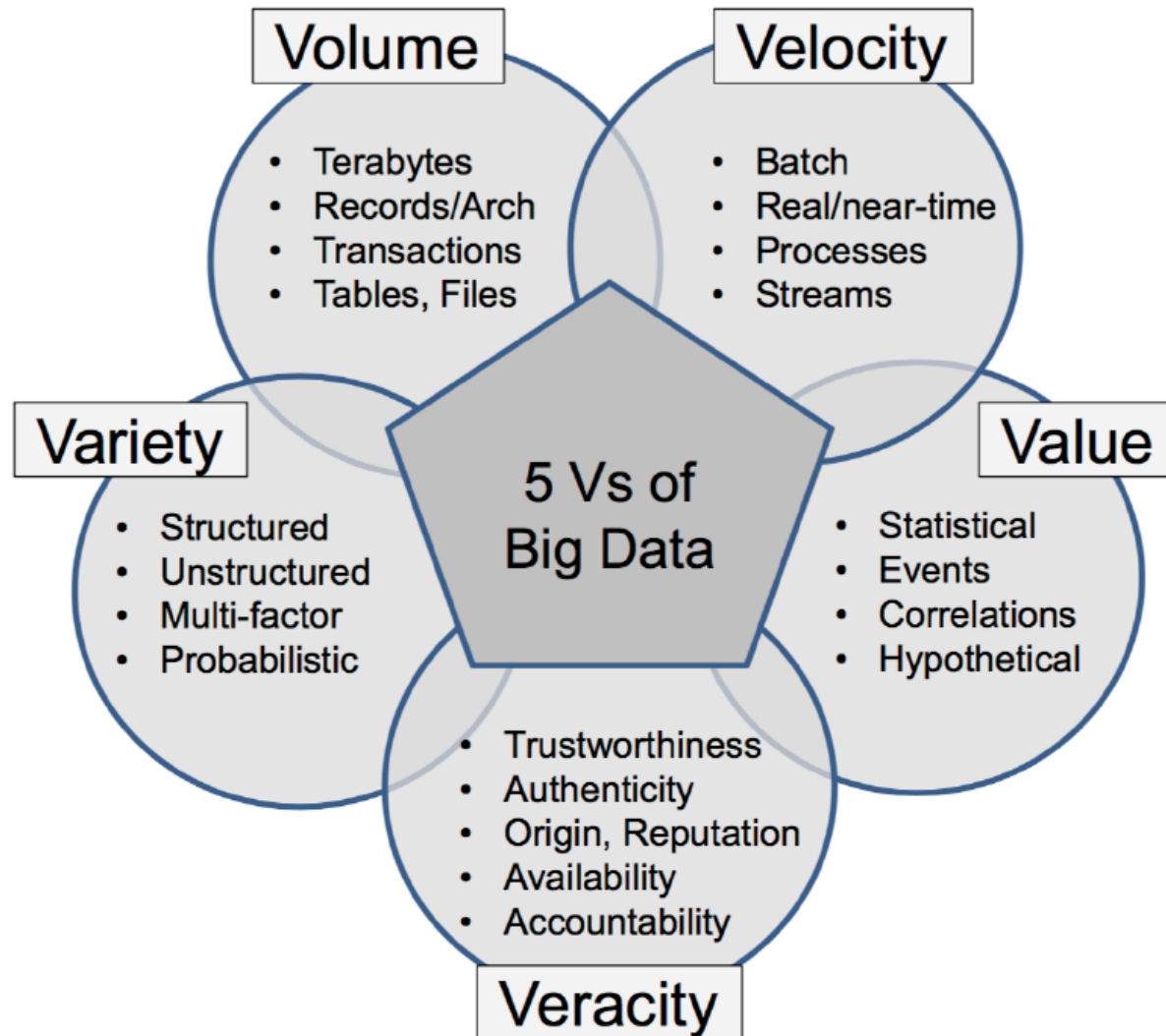
# Big Data

---

- «A new generation of technologies and architectures designed to economically extract value from very large volumes of a wide variety of data by enabling high-velocity capture, discovery, and/or analysis»
- Technologies
  - Techniques for analyzing data, such as machine learning
  - Big Data technologies, like business intelligence, cloud computing and databases
  - Visualization, such as charts, graphs and other displays

# 5V de Big Data

---



# Plan du cours

---

- Introduction
- Calcul distribué sur cluster et grille
- Le Cloud
- Conteneurs (Docker)
- Big Data : introduction
- **EcolInfo : quelques mots**
- Conclusions

# Impacts environnementaux

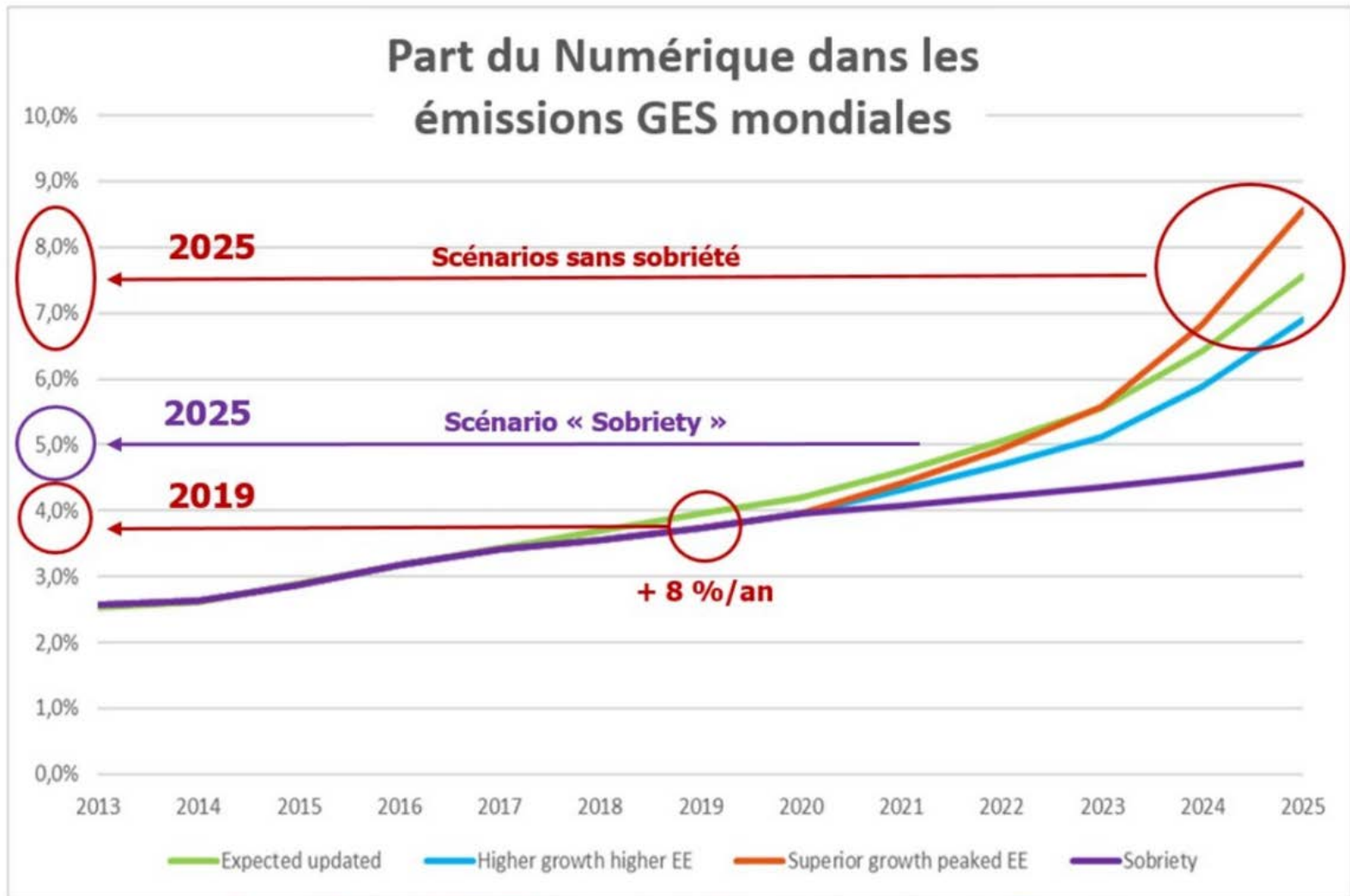


Figure 1 : Evolution 2013-2025 de la part du Numérique dans les émissions de GES mondiales  
[Source: « Lean ICT – Pour une sobriété numérique » (The Shift Project, 2018)]

# Exemple de consommation

---

- Consommation 2021 Centre de Calcul Strasbourg
  - Cloud Computing : 51628 kW / an pour 1024 cœurs (50 kW/an/cœur)
  - Stockage : 34147 kW / an pour 768 To -> soit 44 kW / an / To
- En France ~ 60 gCO<sub>2</sub>e / kW
  - ~ 3 kg de CO<sub>2</sub> / cœur / an
  - ~ 2,7 kg de CO<sub>2</sub> / To / an
- Il faut ajouter 30 % pour la climatisation et l'environnement de la salle à ces chiffres
- Crédits et remerciements : Jérôme Pansanel

# Exemple matériel

---

Etude d'Apple en 2019 sur les portables MacBook 16 pouces stockage 521 Go, fréquence 2,6 GHz.

- La phase d'utilisation en première main est considérée à 4 ans.
- l'empreinte Carbone est de 394 kgCO<sub>2</sub>e
- la fabrication constituant alors 75%
- le transport 5%, l'utilisation 19%
- la fin de vie le 1% restant.

Il faudrait utiliser cet ordinateur **4 fois plus longtemps** pour que la part de l'utilisation soit du même ordre que la fabrication.

Etude similaire chez Dell en 2018 avec des conclusions similaires

# EcoInfo

---

- <https://ecoinfo.cnrs.fr/>
- Réduire les impacts environnementaux et sociétaux négatifs des technologies du numérique
- Agir vers la sobriété numérique
  - Réfléchir avant d'agir : réfléchir à ses besoins logiciels et matériels
  - Refuser (ne pas) : ne pas produire/conserver de données inutiles, etc ...
  - Réduire et introduire de la sobriété dans les usages, les achats, la consommation
  - Réutiliser : favoriser systématiquement le réemploi de matériel fonctionnel (le coût environnemental étant principalement lié à la fabrication des objets technologiques)
  - Recycler en utilisant les filières officielles
  - Acheter de préférence auprès de constructeurs engagés dans une démarche éco-responsable
  - Partager : mutualiser le matériel et les licences logicielles, ouvrir ses données et son code
  - Élargir sa vision des impacts et s'interroger sur la pertinence des choix et des décisions au niveau de son groupe de travail, voire de la société

# Résumé et conclusions

---

- De plus en plus de données à traiter
  - Appel au calcul distribué sur clusters, grilles et clouds
  - Bien poser le problème
  - Optimisations toujours possibles
  - Reproductibilité
- Cloud
  - Calcul distribué
  - Stockage
  - Machines virtuelles et conteneurs
  - Services à la demande
- Virtualisation et Conteneurisation

# Remerciements

---

- Jérôme Pansanel

- IPHC, Strasbourg

- Tristan Glatard

- Concordia University, Montreal

- <https://github.com/glatard/big-data-analytics-course>

- FranceGrilles

---

# Merci pour votre attention!

---

Des questions ?

# Quelques liens supplémentaires

---

- <https://towardsdatascience.com/a-neanderthals-guide-to-apache-spark-in-python-9ef1f156d427>
- <https://spark.apache.org/docs/latest/quick-start.html>
- <https://spark.apache.org/docs/latest/mllib-clustering.html>
- <https://www.edureka.co/blog/mapreduce-tutorial/>
- <https://data-flair.training/blogs/how-apache-spark-works/>

# Quizz

---

- What is the difference between HPC and HTC ?
- What is a cluster/batch manager ?
- Name three differences between Grids and Clouds
- What is OpenStack ?
- What is Heat ?
- What is Hadoop? Name three Hadoop components
- Name one main characteristic of a Map task/function
- What is EOSC? What about Gaia-X?
- What is a container? Why can it be interesting for you ?