# Lecture One

Introduction to the Course and to Cloud Computing

# Introduction and course overview

- What is the Cloud
- Virtualization and Containers
- What is the Cloud Computing architecture and service model
- Data Management in the Cloud
- Cloud Security and economy

...and examples based on public clouds and scientific use cases.

# Cloud Computing: an historical view

Cloud Computing concept is the result of the evolution of the computing concept driven by the technology improvements and by users requirements.

**Cloud Computing**

Utility computing

Grid Computing

Distributed computing

Centralized Computing (mainframe)

HPC HTC

Cluster computing

# What is computing

“Computing is the process of using computer technology to complete a given goal-oriented task. [...] Computing may encompass the design and development of software and hardware systems for a broad range of purposes”

(Association of Computing Machinery, 2005)

# Computing as a “numerical laboratory”

Each scientific instrument is critically **dependent on computing** for sensor control, data processing, international collaboration, and access.

**Computational modeling and data analytics** are applicable to all areas of science and engineering

**Capture and analyze the torrent of experimental data** being produced by a new generation of scientific instruments

# Distributed Computing

From a single computer to a "network" of collaborating systems.

*"A distributed system is a collection of autonomous computers that are interconnected with each other and cooperate, thereby sharing resources such as printers and databases"* (C. Leopold)

We first introduce the role of the network as a glue of multiple resources

# Distributed Computing Motivations

Some applications are **inherently distributed problems** (they are solved most easily using the means of distributed computing)

Computing intensive problems where communications is limited (**High Throughput Computing)**

**Data Intensive problems**: computing task deal with a large amount or large size of data.

Distributed computing allows for "**scavenging**." By integrating the computers into a distributed system, the excess computing power can be made available to other users or applications (e.g. Condor)

Robustness: no single point of failure.

more….

# Distributed computing properties

**Fault tolerance**

- if a node fails the whole system still work
- each node play a partial role (partial inputs and outputs)
- check node status

**Resource sharing**

**Load Sharing and balance**

- to distribute computing on different nodes to share loading to the whole system

Easy to expand (**scalability**)

Improve **performance**

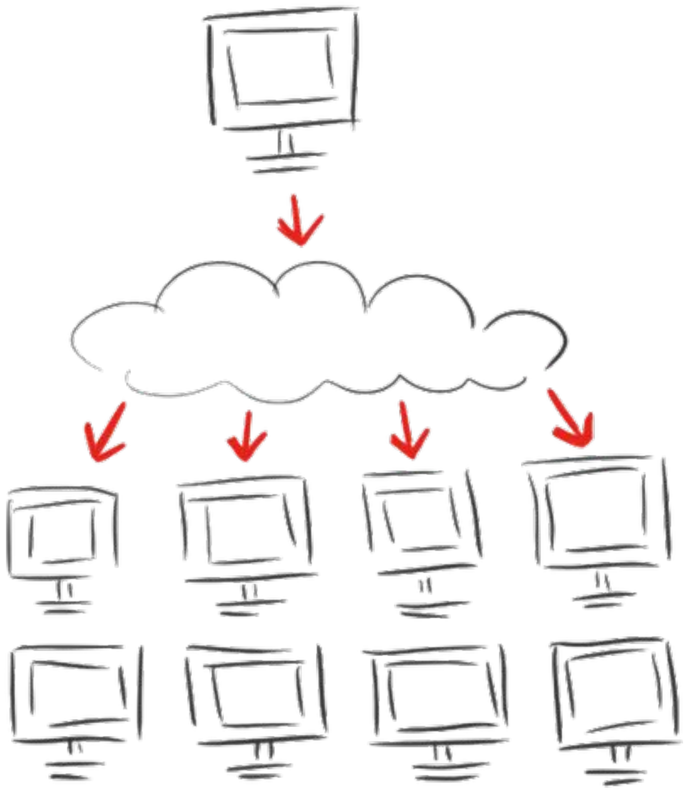
# Distributed computing Architecture

*“interconnect processes running on different CPUs with some sort of communication system.”*

**client-server:** resource management centralized at a server

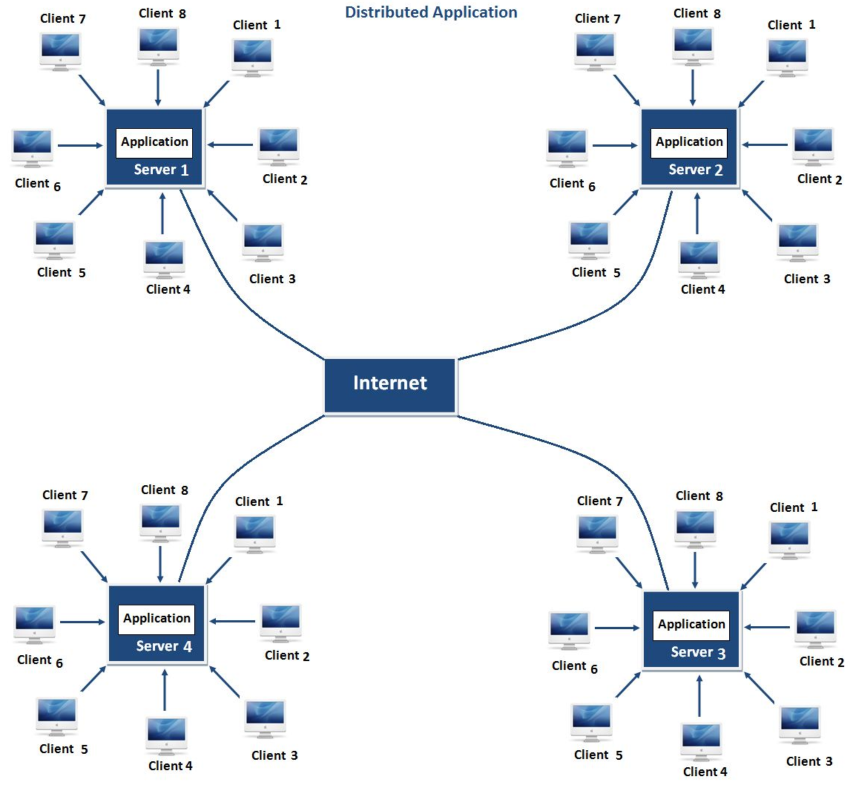
**3-Tier architecture**: move the client intelligence to a middle tier to simplify application development.

**Peer-to-Peer**: responsibilities are uniformly divided among all machines, known as peers that serves both as client and servers



# Distributed Applications

*“A distributed application is software that is executed or run on multiple computers within a network. These applications interact in order to achieve a specific goal or task.”*



# Examples of Distributed Systems

SETI@Home, Folding@Home…

Peer-to-Peer networks

High Availability Systems

Distributed databases

High Throughput Computing

Parallel Computing

...even the World Wide Web is a distributed system.

# BigData distributed computing and Hadoop

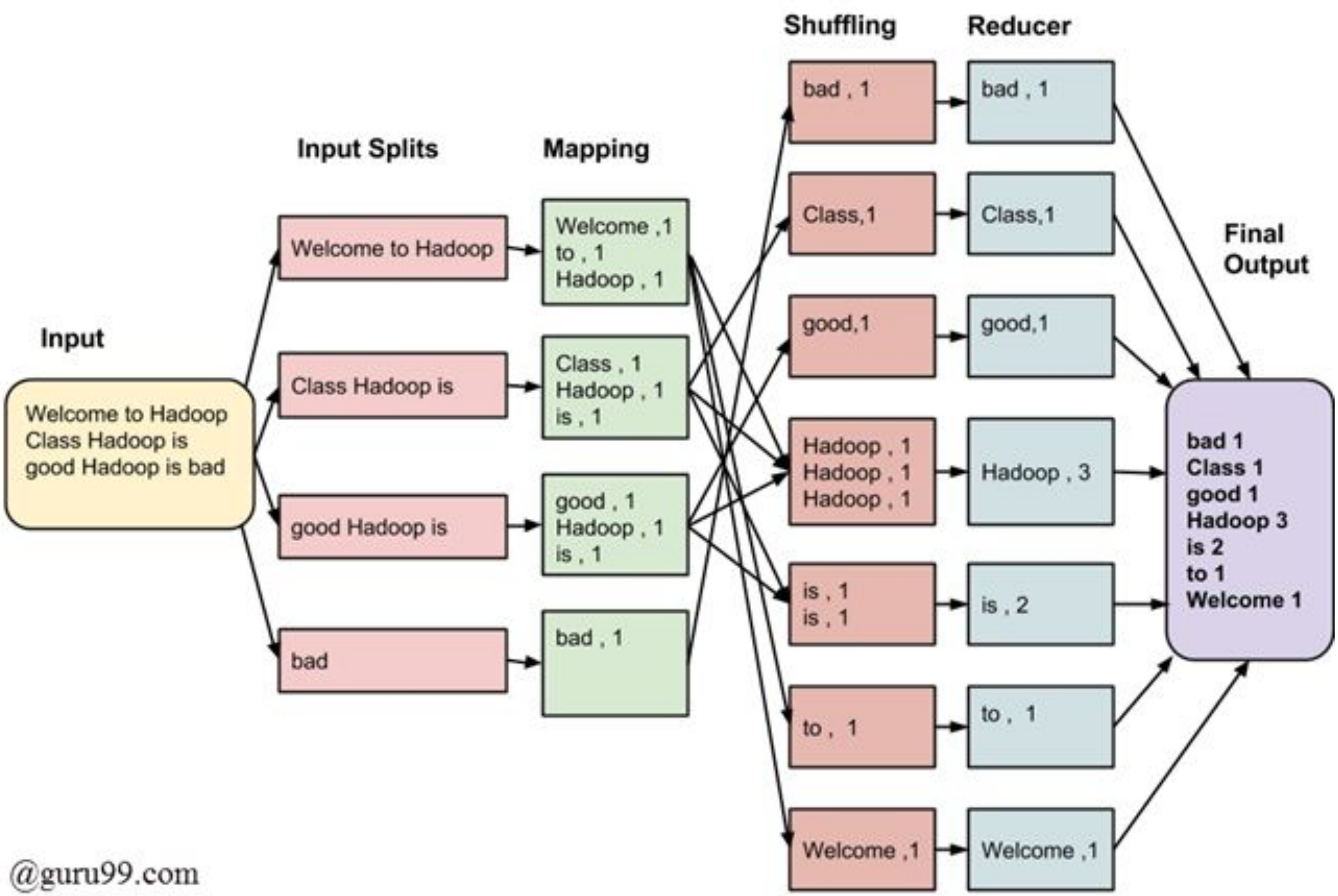
Apache Hadoop system implements a MapReduce model for data analytics

A distributed file system (HDFS) manages large numbers of large files, distributed (with block replication) across the storage of multiple resources

Tools for high-level programming model for the two-phase MapReduce model (e.g. PIG)

Can be coupled with streaming data (Storm and Flume), graph (Giraph), and relational data (Sqoop) support, tools (such as Mahout) for classification, recommendation, and prediction via supervised and unsupervised learning.

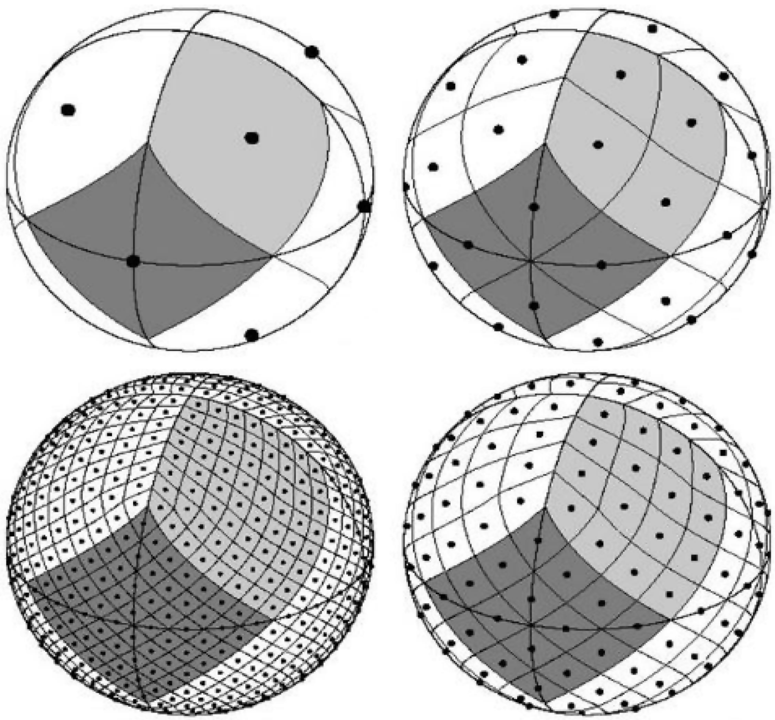
# Example of MapReduce

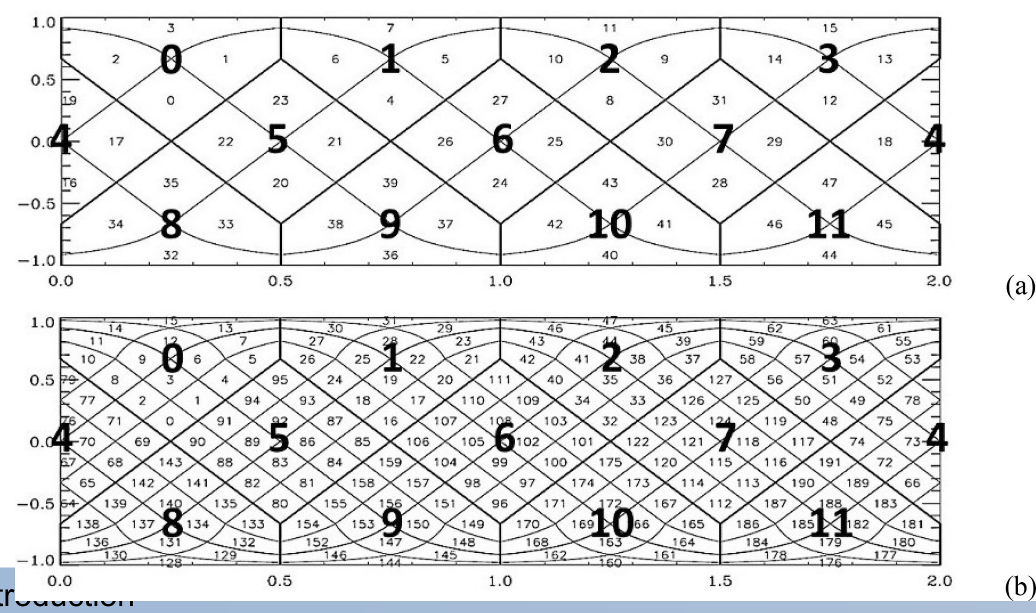



https://www.guru99.com/introduction-to-mapreduce.html

# Distributed applications in Astronomy using Hadoop

Hierarchical Equal Area iso-Latitude Pixelization (HEALPix).





# Hadoop & Spark, « cross-match » of source catalogues

André Schaaff, François-Xavier Pineau
CDS, Centre de Données astronomiques de Strasbourg
Noémie Wali
UTBM, Université de technologie de Belfort-Montbéliard

Special thanks to Julien Nauroy, Université de Paris Sud

**IVOA Cape Town, GWS session 2**

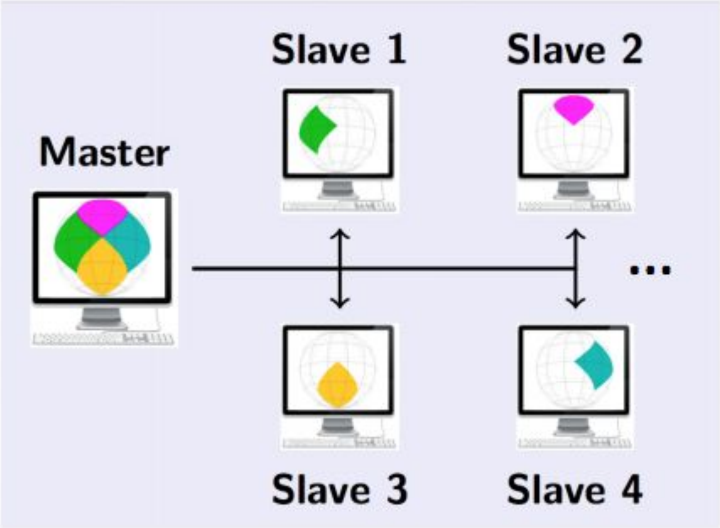
Centre de Données Astronomiques de Strasbourg

IVOA

Asterics
Astronomy ESFRI & Research Infrastructure Cluster

# Distribution ?

- With Hadoop / Spark, the data is distributed over several nodes
- Distribution ?
- How to optimise it ?

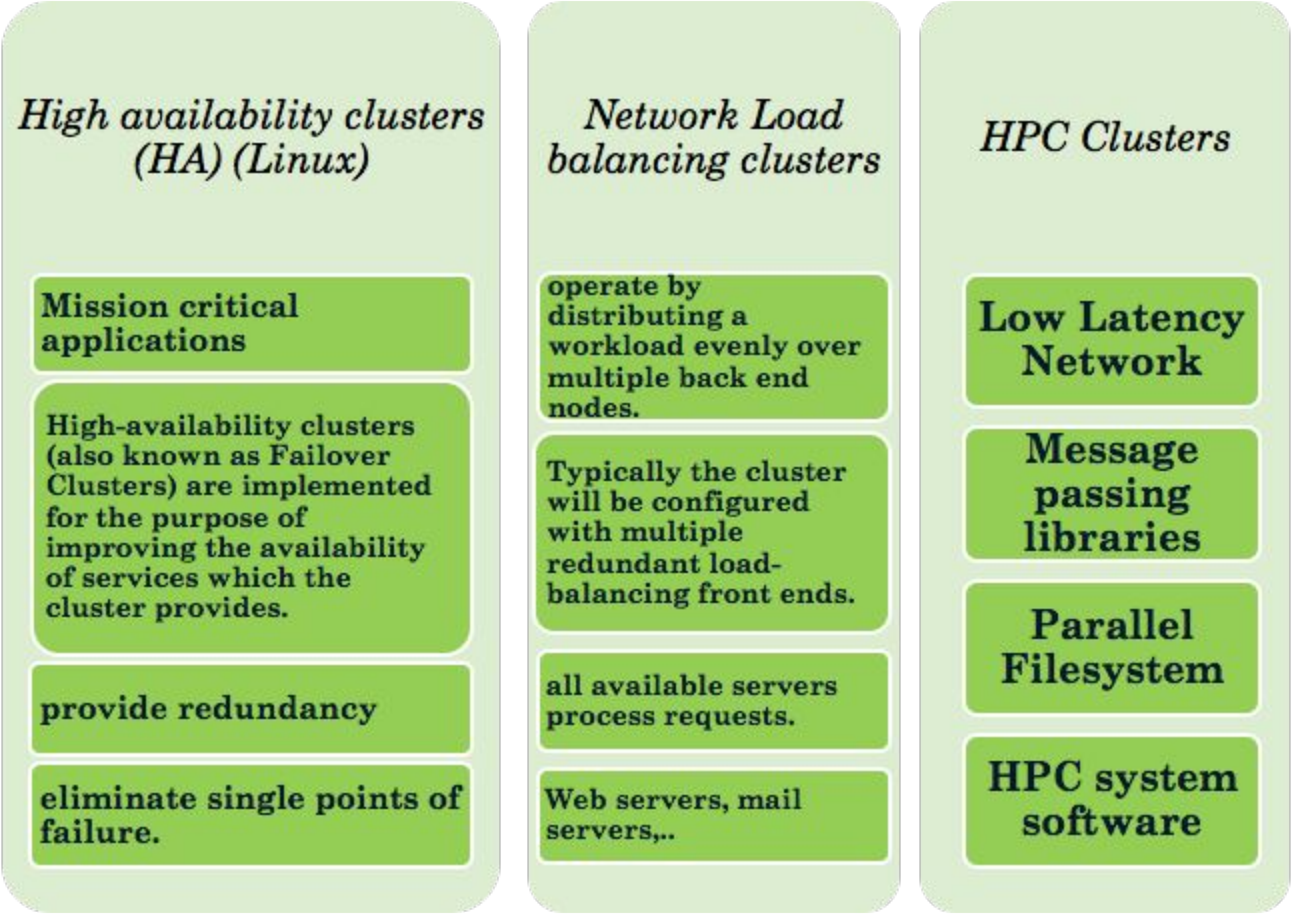


# Cluster Computing and HPC

A **computer cluster** is a group of *linked* computers, working together closely so that in many respects they form a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks.

Clusters are usually deployed to **improve performance** and/or **availability** over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

# Cluster Classification



# HPC clusters components

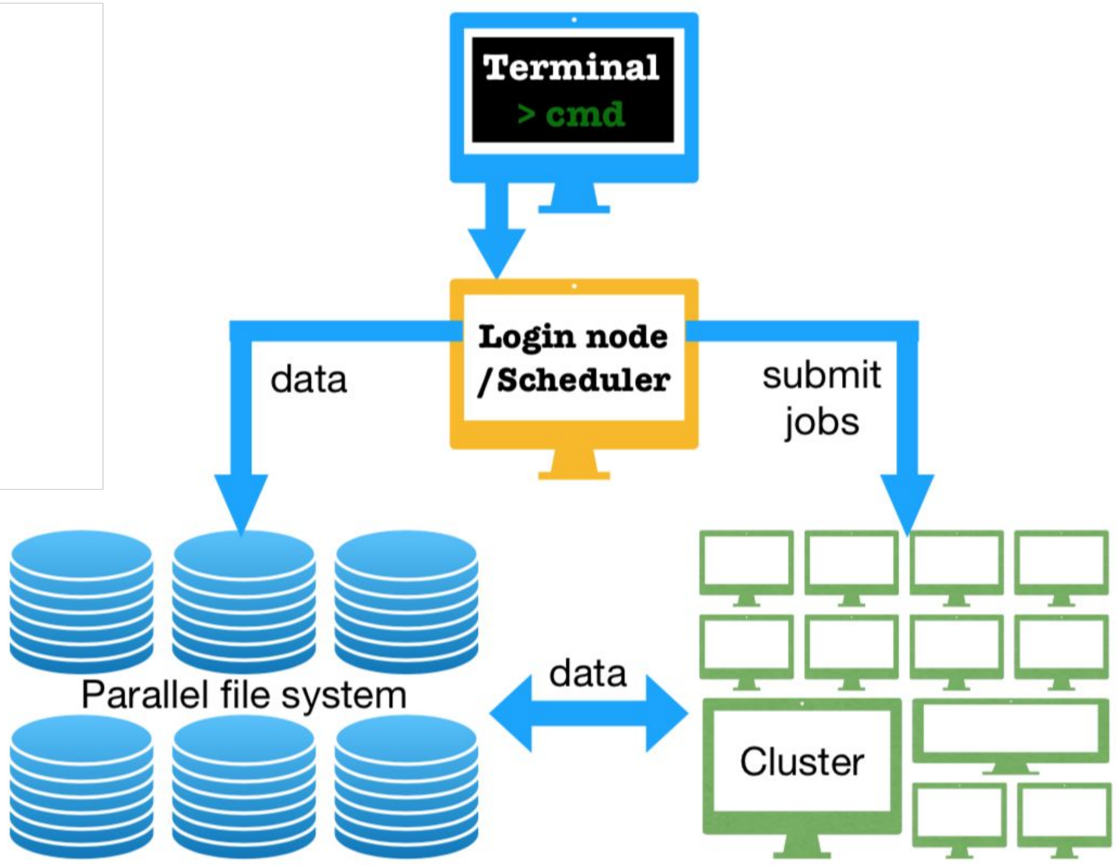
Login node

Controller node

Computing Nodes

Parallel filesystem

System software



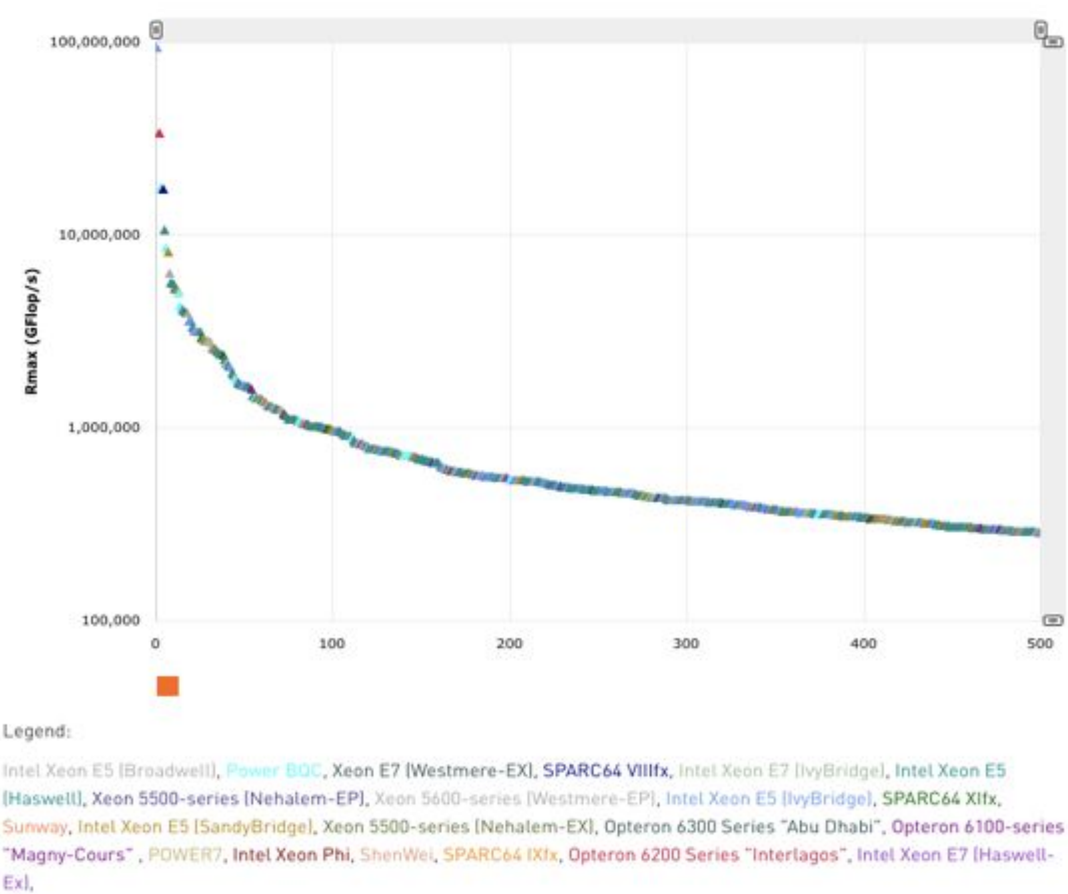
# Benefit of HPC clusters

- Cost-effective
  - Much cheaper than a super-computer with the same amount of computing power!
  - When the supercomputer crashes, everything crashes, when a single/few nodes in HPC fail, cluster continues to function.
- Highly scalable
  - Multi-user shared environment: not everyone needs all the computing power all the time.
  - higher utilization: can accommodate variety of workloads (#CPUs, memory etc), at the same time.
  - Can be expanded, partitioned  or shrunk, as needed.

# HPC clusters today

HPC clusters are heterogeneous environments where the computing power is  given by CPU and Accelerators

FUGAKU - 48 cores Armv8.2 2.2GHz  System, Tofu Interconnect, 158976 nodes, 7299072 cores, **415530 TFlop/s**



# High Performance Data Analysis

The ability of increasingly powerful HPC systems to **run data-intensive problems** at larger scale, at higher resolution, and with more elements (e.g., inclusion of the carbon cycle in climate ensemble models)

The proliferation of larger, more complex scientific instruments and sensor networks, from "smart" power grids to the Large Hadron Collider and Square Kilometer Array.

The growth of stochastic modeling, parametric modeling  and other iterative problem-solving methods, whose cumulative results produce large data volumes.

The availability of **newer advanced analytics methods and tools**: MapReduce/Hadoop, graph analytics (NVIDIA IndeX), semantic analysis, knowledge discovery algorithms (IBM Watson), COMPS and pyCOMS, and more

The escalating need to perform advanced analytics in **near-real time** a need that is causing a new wave of commercial firms to adopt HPC for the first time

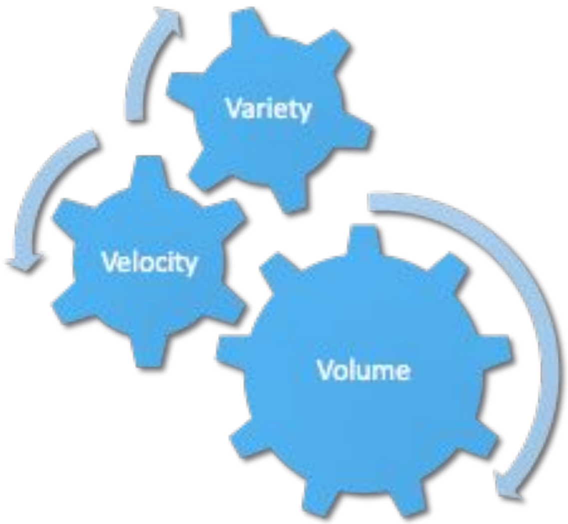
# HPC Clusters usage

**Complexity**. HPC technology allows scientist to aim more complex, intelligent questions at their data infrastructures.

**Time to value**. Science faces ever-shortening innovation and production cycles. Analytics (including Hadoop and Spark) is moving from batch processing toward low-latency, interactive capabilities.

**Variability**.  “deep” vs “Wide”

“large amount of data” vs “many variables”



# How to use a cluster

Batch systems: a job is executed via a scheduler that optimize the use of cluster resources

Applications must be executed using a shell script to load the proper application environment (libraries, paths, tools)

Not suitable for interactive jobs (not completely true we can use so tricks to make it works)

Batch is organized in queue with a limited computing time (application snapshot and restart capabilities)

Complex filesystem structure: home, scratch, data

# Grid Computing

“a single seamless computational environment in which cycles, communication, and data are shared, and in which the workstation across the continent is no less than one down the hall"

“wide-area environment that transparently consists of workstations, personal computers, graphic rendering engines, supercomputers and non-traditional devices: e.g., TVs, toasters, etc."

“[**framework** for] flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources"

“collection of geographically separated resources (people, computers, instruments, databases) connected by a high speed network [...distinguished by...] a software layer, often called middleware, which transforms a collection of independent resources into a single, coherent, virtual machine”

# Why do we need Grid computing?

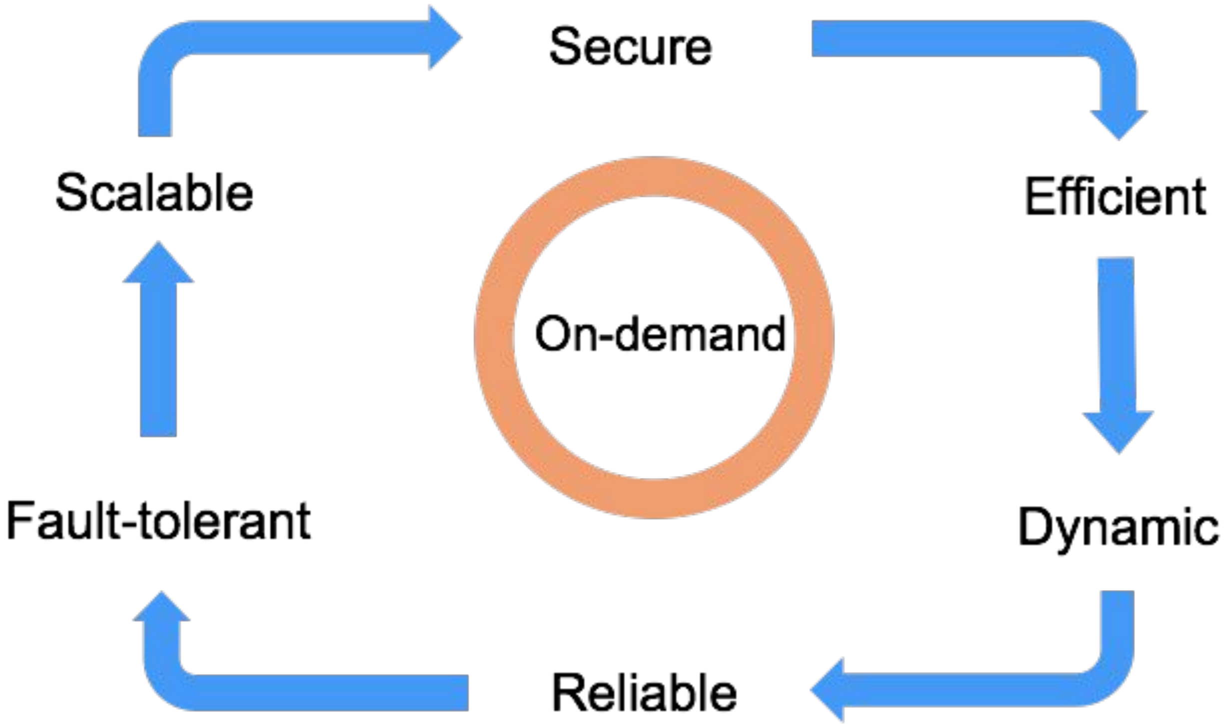
- Going further in scientific knowledge
  - New high sensitivity sensors and instruments
  - Globally distributed collaborations
- Delocalized knowledge
  - Scientific and technical knowledge is “distributed”
  - Laboratories are distributed
  - Scientific data are distributed
- Exploiting under utilized resources.

# Virtual Organizations

The size and/or complexity of the problem requires that people in several organizations collaborate and share computing resources, data, instruments

VIRTUAL ORGANIZATIONS

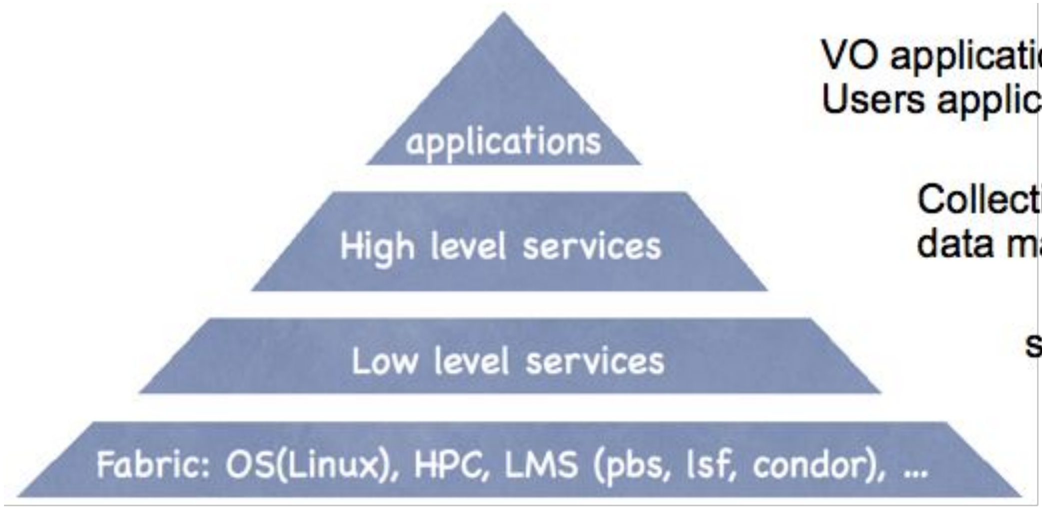
# Grid Concepts



# Grid Middleware

It's the software layer that glue all the resources

Everything that lies between the OS and the application




VO application layer
Users application layer

Collective services: scheduling,
data management, info sys…

security, jobs, file transfer…

execution,
files storing, DB access…

# Examples of Grid Computing

Globus alliance (Globus Toolkit)

gLite (EGEE middleware)

Unicore (DE)

GridBus

GRIA

*LHC data has been distributed on a tiered architecture based on LHC Computational Grid (gLite) and processed using the LHC Grid.*

# Grid Limitations

Very **Rigid environment**: all the resources must be installed, maintained and monitored homogeneously.

Useful for applications that requires an **HTC environment,** but a high level of complexity is introduced to use it efficiently

Licensing problems across different  domains

Implementation limits due to the **middleware** used.

Political challenges associated to resource sharing

# Utility Computing

It is a theoretical concept, and CC implements this concept in practice

“*It is a service provisioning model in which a service provider makes computing resources and infrastructure available to customers and charges them for specific usage rather than a flat rate*” (on-demand)

Low or no initial cost to get a resource (the resource is essentially rented)

Pay-per-use model
maximize the efficient use of resources minimizing costs

# Main Concept of Utility Computing

1. Pay-per-use Pricing Business Model
2. Optimize resource utilization
3. Outsourcing
4. “infinite resource availability”
5. Access to applications or libraries
6. Automation

# Utility Computing

The principle of utility computing is very simple: One company pays another company for servicing. The services include software rental, data storage space, use of applications or access to computer processing power. It all depends on what the client wants and what the company can offer.

# Utility Computing in practice

Data backup,

Data Security

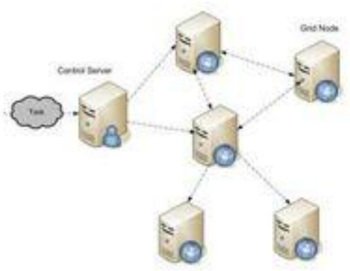
Partners competences

Defining a SLA

Getting Value from charge back

# Computing evolution

**Grid Computing**
- Solving large problems with parallel computing
- Made mainstream by Globus Alliance



**Utility Computing**
- Offering computing resources as a metered service
- Introduced in late 1990s

**Software as a Service**
- Network-based subscription to applications
- Gained momentum in 2001


**Cloud Computing**
- Next-Generation Internet computing
- Next Generation Data Centers

# End of lecture 1

Next time we will see the basic of CC and CC architecture.