

Introduction to Cloud COMPUTING

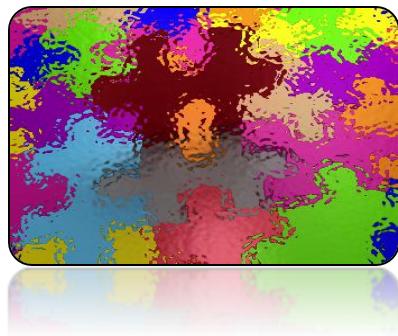
Giuliano Taffoni- I.N.A.F.



“Introduction to Cloud Computing”

2023 @ Università di Trieste

Outline



Intro to Cloud
Computing lectures



Computing and
Cloud Computing

What to expect from this course

- ✓ 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.

More in details

Introduction - Computing and computing models; Distributed Computing; Cluster computing and HPC; Grid Computing; Utility computing; Towards Cloud computing.

Cloud Computing Architecture - Properties, Characteristics and Disadvantages; Role of Open Standards; Cloud stacks; Service models; Deployment models; Cloud types and usage.

Virtualization - Virtual Machines; Cloud and Virtual Machines;

Containers – Container and container images;

Resource Management - AWS IaaS and Examples;

Data Management - Store and access data; scalability and usability; database and data store; large scale data processing;

Cloud Security - Infrastructure Security; Data Security (storage); Identity and access; Access Control, Trust, Reputation, Risk;

Orchestration – K8s basic introduction

| More in details - Lectures

Are based on slides and examples: the lectures introduce basic and advanced concepts (we do expect you to know the basic and understand the advanced).

Questions: you ask what you want whenever you want

Discussions: you ask to discuss much deeper

Office hours: each of us will be available weekly an hour (we'll communicate which one). However, we are ready for chatting, Q&A or alike upon appointment.

More in details...Labs

Virtual Machines – Create a cluster of VMs, Manage images, disks network, benchmark the resources;

What you need: VirtualBox hypervisor installed in your system.

Containers – how to use a container, build a container image, container networking, volumes and orchestration

What you need: Docker desktop (but also podman...)

What do we expect?

Follow the tutorial, experiment during the tutorial and later at home...the exam will be easier then...

| 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)

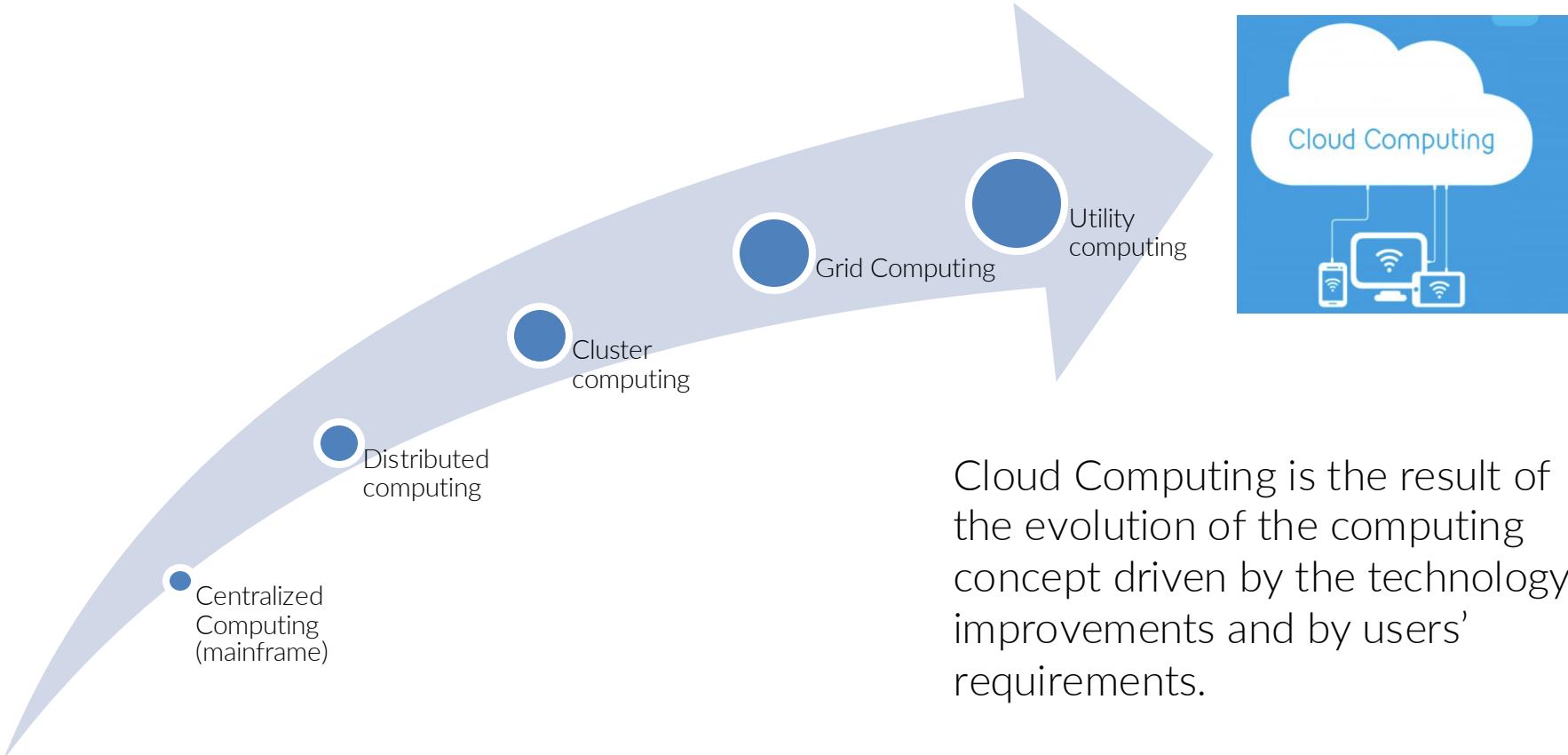
Modern computing

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

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

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

From Distributed to Cloud



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)

The role of the network as a glue of multiple resources.

Distributed Computing

Some applications are inherently **distributed problems** (they are solved most easily using the means of distributed computing) *EXAMPLE: EVERY COMPUTER HAVE IS OWN TASK*

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.

*ABILITY OF USING ALL CYCLE OF
CPU FOR DOING TASKS => VERY USED
IN THE 90s*

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

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 – between users and between sites

Load Sharing and balance - to distribute computing on different nodes to share loading to the whole system

Scalability - Easy to expand

Performance - More resources involved

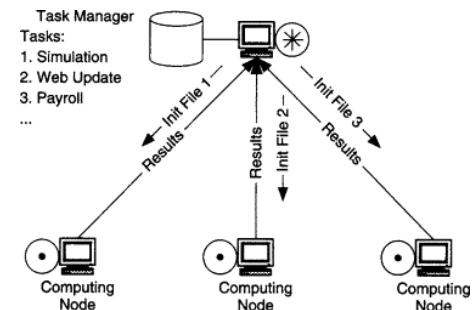
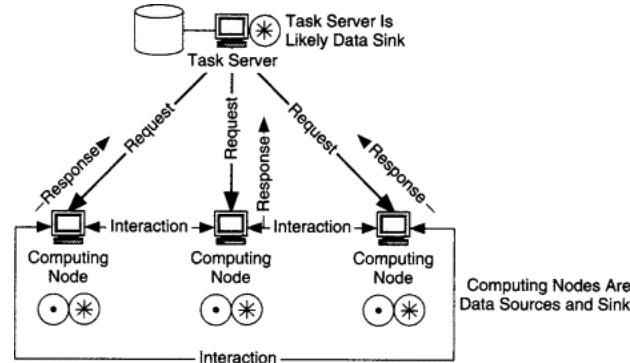
DD Architecture

"interconnect processes running on different CPUs with some sort of communication system."

client-server: resource management centralized at a server

*ALL LOGIC ARE ON THE SERVER
THE CLIENT JUST CONNECT AND
ASK FOR THE ANSWER*
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 serve both as client and servers



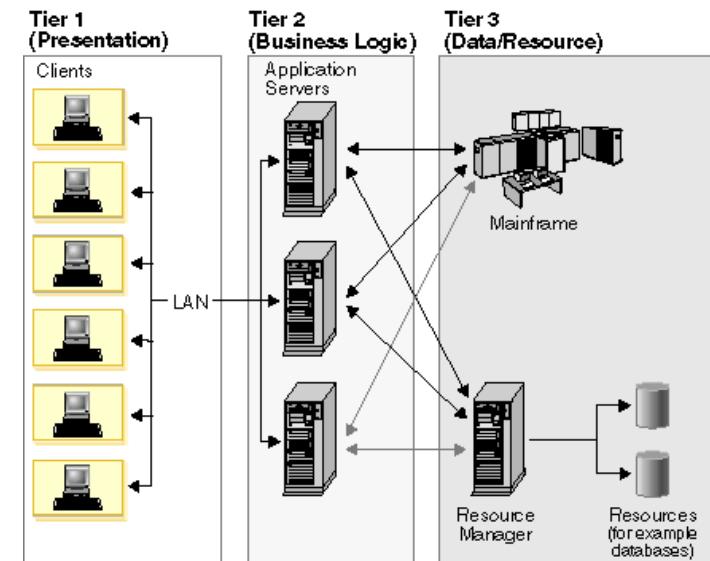
DD Architecture

"interconnect processes running on different CPUs with some sort of communication system."

client server resource management
SPLITTING THE CAPACITY
centralized to decentralized
OF THE SERVER
THE DATA ARE ON ANOTHER SYSTEMS

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 serve both as client and servers



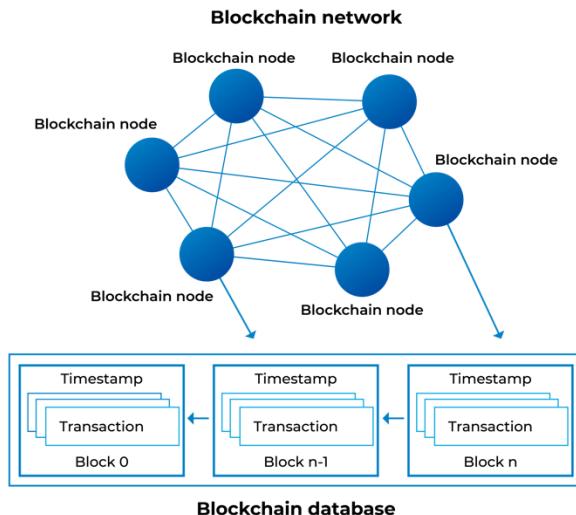
DD Architecture

“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 serve both as client and servers

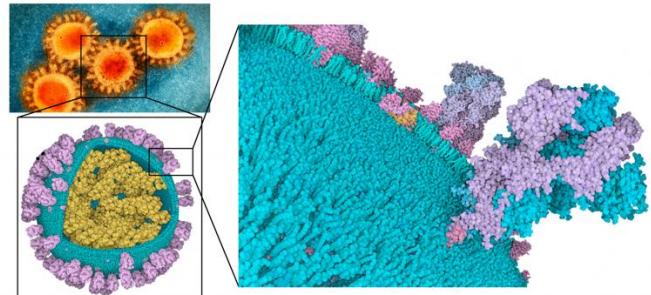


Distributed applications

OpenPandemics - Covid19

High Availability Systems
Distributed databases
High Throughput Computing

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



Apache hadoop

Apache Hadoop system implements a distributed scalable computing 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.



MapReduce Example

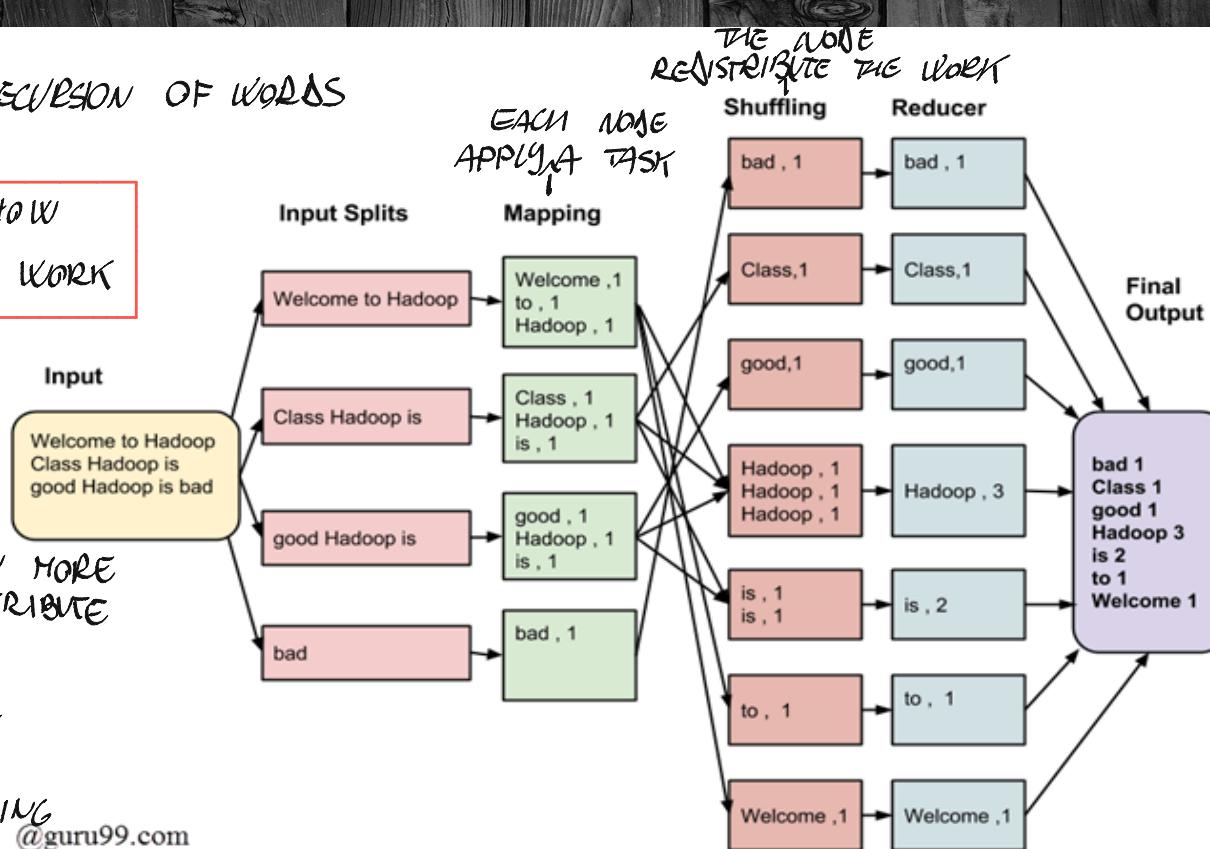
• COUNTING RECURSION OF WORDS

THIS IS HOW
PARALLEL
PROGRAMMING WORK

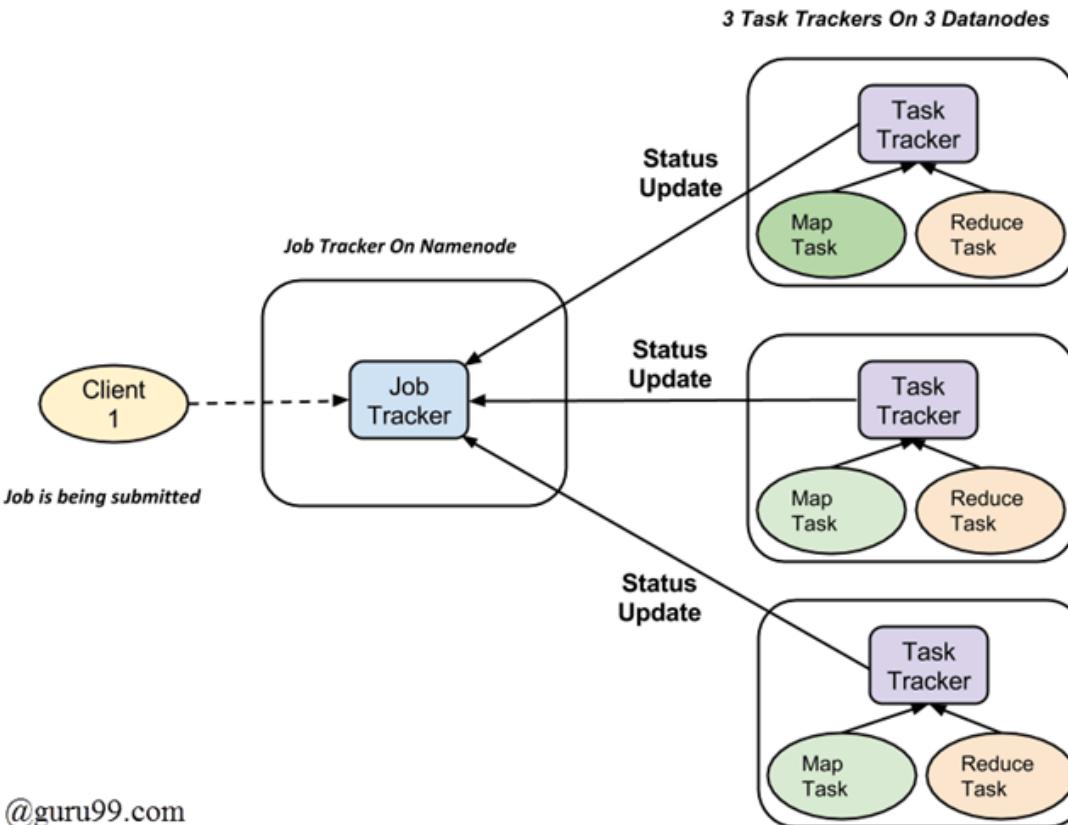
SPLITTING ALLOW MORE
EASILY TO DISTRIBUTE
WORK.

BUT, TOO SMALL
AND IT BECOME
USELESS, INCREASING
WORK TIME

@guru99.com



MapReduce work



multiple tasks onto multiple data nodes

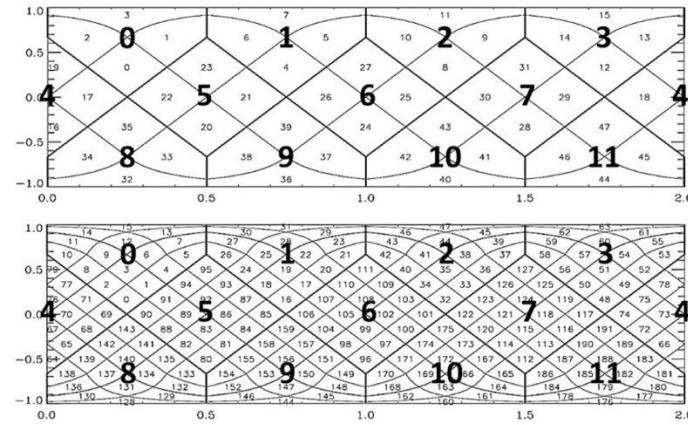
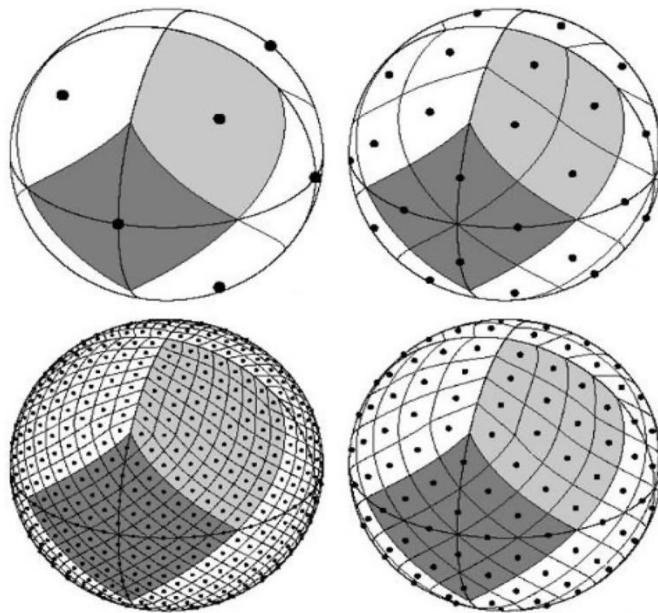
job tracker - coordinate and scheduling

Task tracker sends the progress and 'heartbeat'

job tracker keeps track of the overall progress. In the event of task failure, the job tracker can **reschedule** it on a different task tracker.

Hadoop in Astronomy

Hierarchical Equal Area iso-Latitude Pixelization (HEALPix).



(a)

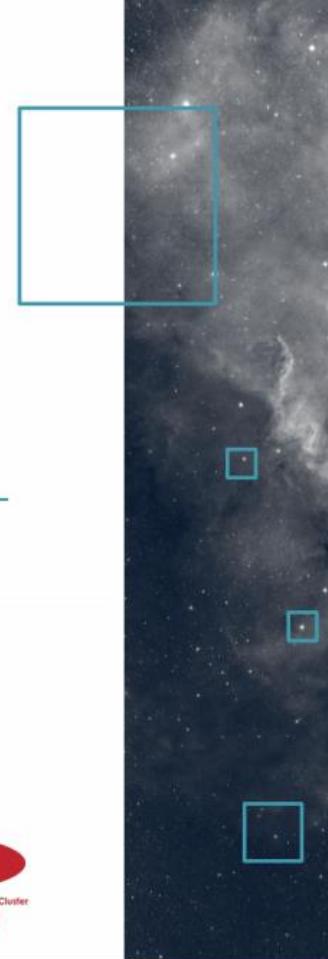
(b)

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

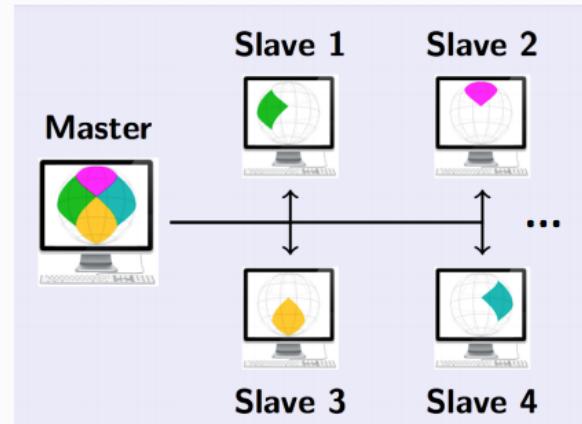


Data distribution and processing

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



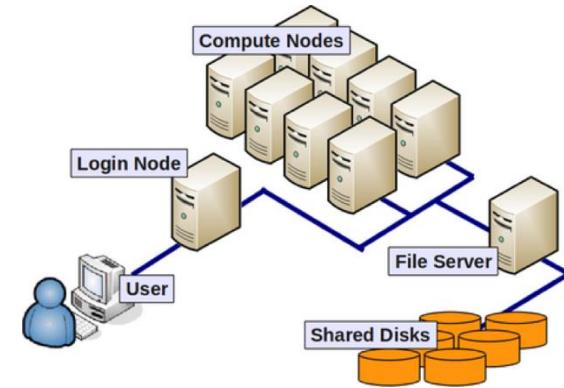
EACH NODE NEED TO
HAVE THE SAME AMOUNT
OF WORK



Cluster Computing

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

I NEED
SOME AMOUNT
OF SERVERS
ALWAYS ALIVE
+ SOME IN IDLE
IF A SERVER
DIE,
IT PROVIDE
REDUNDANCY



High availability clusters (HA) (Linux)

Mission critical applications

High-availability clusters (also known as Failover Clusters) are implemented for the purpose of improving the availability of services which the cluster provides.

provide redundancy

eliminate single points of failure.

EXPLAIN BY
HIMSELF

Network Load balancing clusters

operate by distributing a workload evenly over multiple back end nodes.

Typically the cluster will be configured with multiple redundant load-balancing front ends.

all available servers process requests.

Web servers, mail servers,..

HPC Clusters

Low Latency Network

Message passing libraries

Parallel Filesystem

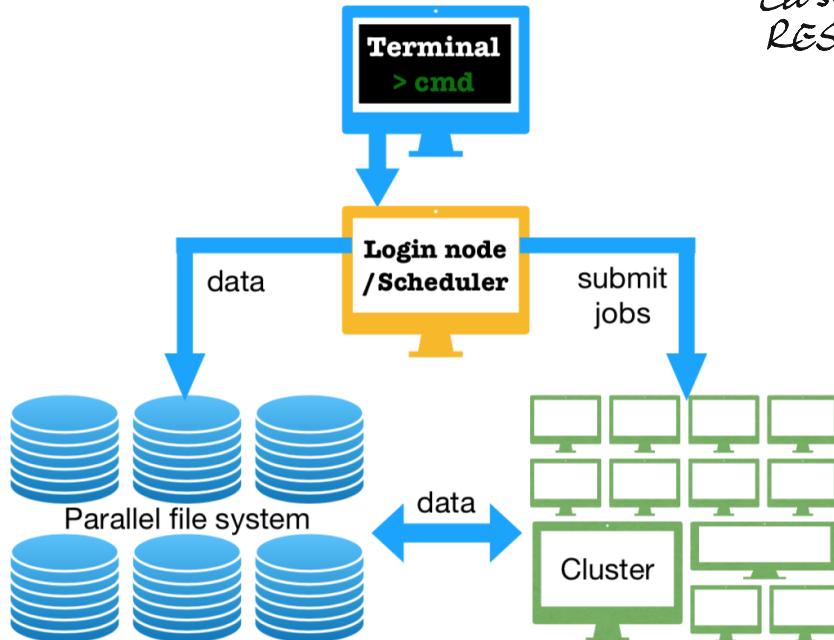
HPC system software

HIGH PERFORMANCE CLUSTER IS A SUPER COMPUTER MADE BY DIFFERENT NODES. IS THE TOUGHEST ONE.

NEED SPECIFIC LIBRARIES

Using a Cluster

LOAD PROAGR
APPLICATION
ENVIRONMENT



A SCHEDULER THAT OPTIMIZE THE USE OF CLUSTER RESOURCES

Batch systems - shell script
- application environment -
Not suitable for interactive
jobs - queue with a limited
computing time - filesystem
structure: home, scratch,

COMPLEX
STRUCTURE

A SOFTWARE HAVE TO BE OPTIMIZE FOR USE THE BENEFIT

Cluster computing is...

Cost-effective

Much cheaper than a super-computer with the same amount of computing power!

Resilient

When the supercomputer crashes, everything crashes, when a single/few nodes in HPC fail, cluster continues to function.

Multi user

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.

Scalable

Can be expanded, partitioned or shrunk, as needed.

HPC Clusters on top500

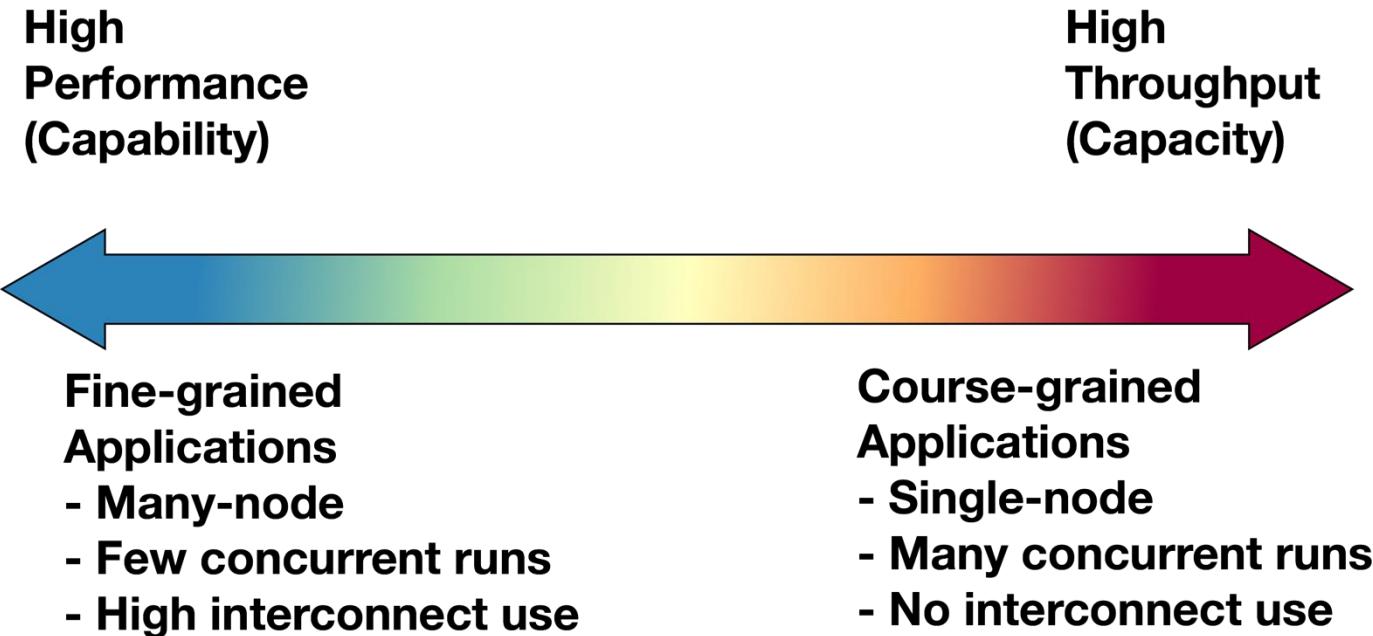
HPC clusters are heterogeneous environments where the computing power is given by CPU and Accelerators...but not all of them ;-)

<https://www.top500.org>

A REAL APPLICATION IS FAR AWAY
FROM WHAT THIS LIST EXPOSE.
THE MEMORY IS THE PRINCIPAL
PROBLEM (LOADING, ACCESSING
AND ECC...) IT NEEDS A GOOD
PROGRAM

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	Leonardo - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404

HPC vs HTC



HTC applications

- divide the problem up into smaller **independent** parts;
- get system to process as many of these small parts as possible in parallel (i.e. at the same time);
↓ INDEPENDENT TASK : NOT COMMUNICATING
- combine the partial results produced by the system to give the overall result.

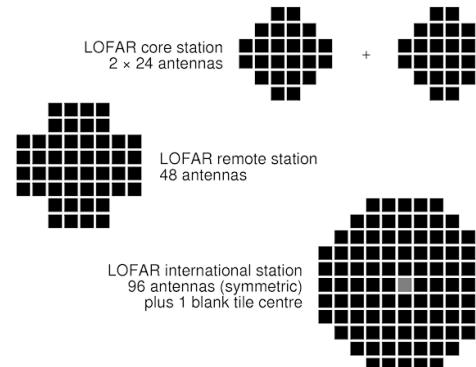
REQUIREMENTS :

Can I partition my data?

Are the small parts independent?

Do you estimate the overhead of partition the data?

HTC example



- Needs to process large datasets (several TBs)
- Datasets can be divided in pieces that can be processed independently.

LOFAR Pipeline parallelism

Each 8 hours observation is 115 MHz to 175 MHz in 371 separate sub-bands. Each sub-band was originally composed of 64 spectral channels, and the initial scan-time was set to 1 second. We need to process 320 observations.

ARE 371 SUB BANDS
INDEPENDENT? YES
 \Rightarrow 371 SPLIT

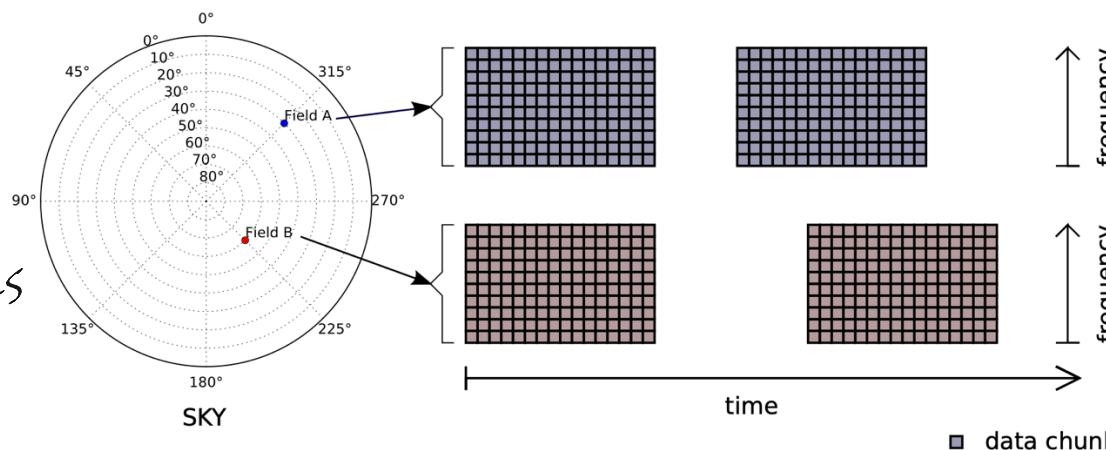
THE 8 HOURS ARE
INDEPENDENT? YES

\Rightarrow SPLIT AGAIN

A GOOD SPLIT

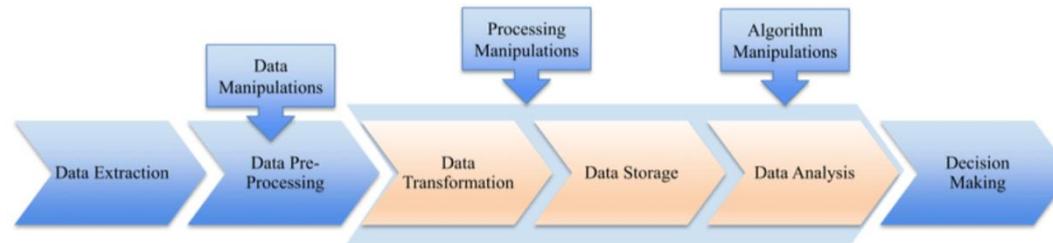
1 WEEK \rightarrow FEW HOURS

SKY+TIME+FREQUENCY data parallelism

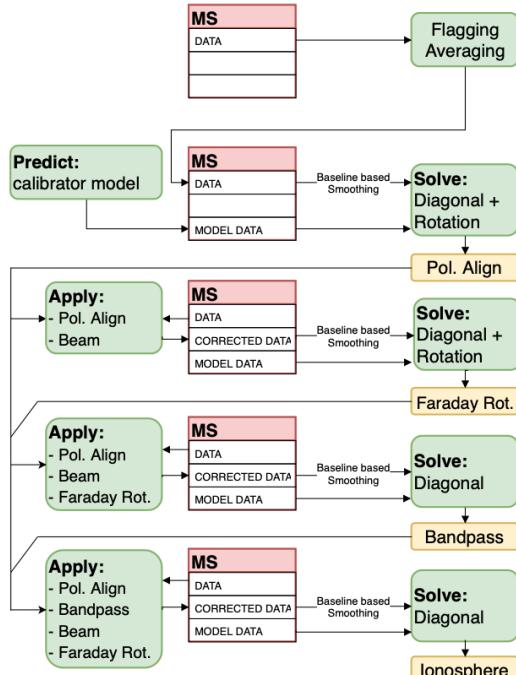


| Data reduction PIPELINE

A pipeline is a set of processes and tools used to collect raw data from multiple sources, analyse it and present the results in an understandable format.



LOFAR Pipeline HTC



Compute Time on single node each observation:

Run on calibrator ~ 2/3 h

Run on target image ~12/24 h

FINAL OUTPUT

~100GB of data in MS format

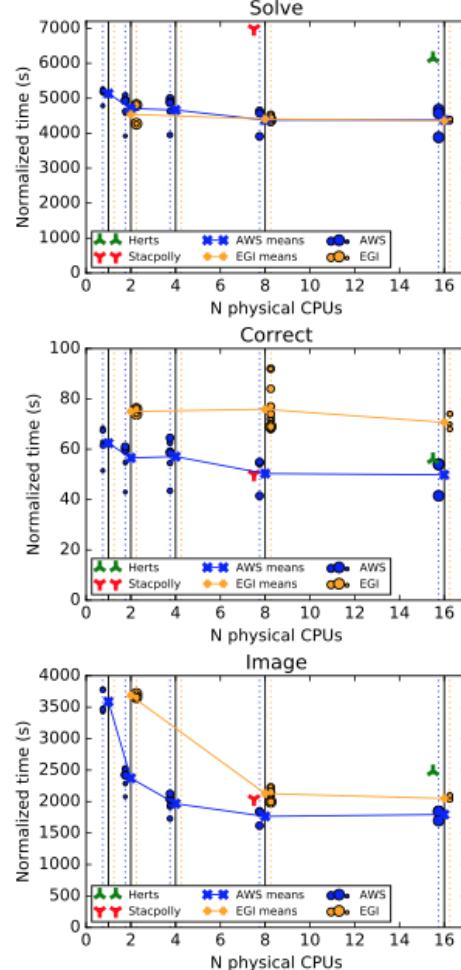
CAN BE SPLITTED IN FREQ (~100)
How much time can we save?

What is the bottle neck?

A LOT OF IO

A LOT OF DATA NEEDS TO BE USED

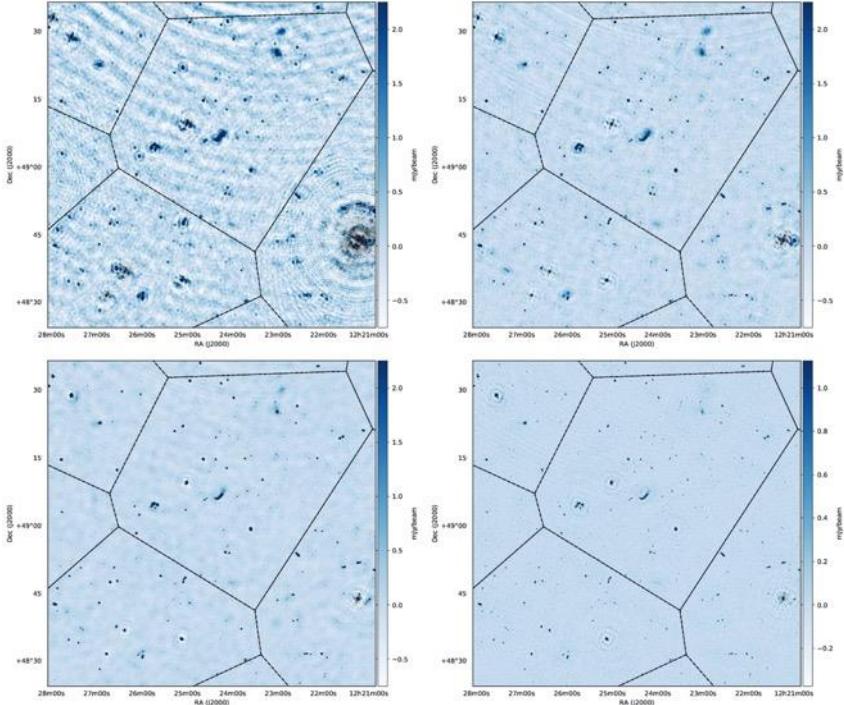
LOFAR Test on single node



High Performance Data Analysis

- The ability of increasingly powerful HTC 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 Kilometre 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

Sizing the problem



Compute Time on single node:

- Run ~8/12 g no snapshots no

FINAL OUTPUT

- 10TBs of data

Sizing the problem

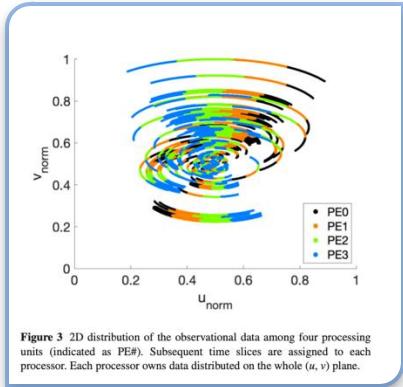
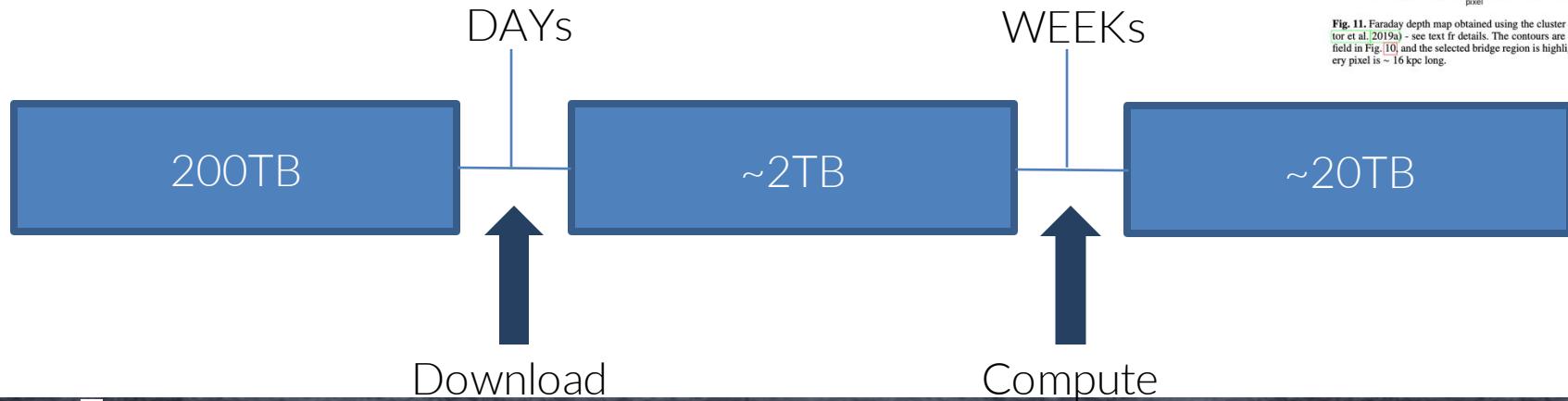
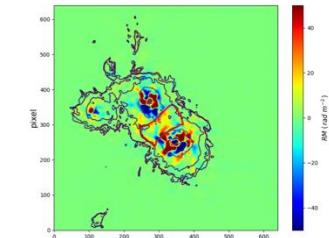
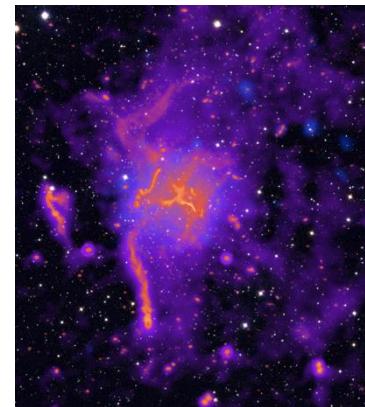


Figure 3 2D distribution of the observational data among four processing units (indicated as PE#). Subsequent time slices are assigned to each processor. Each processor owns data distributed on the whole (u, v) plane.



Future challenges



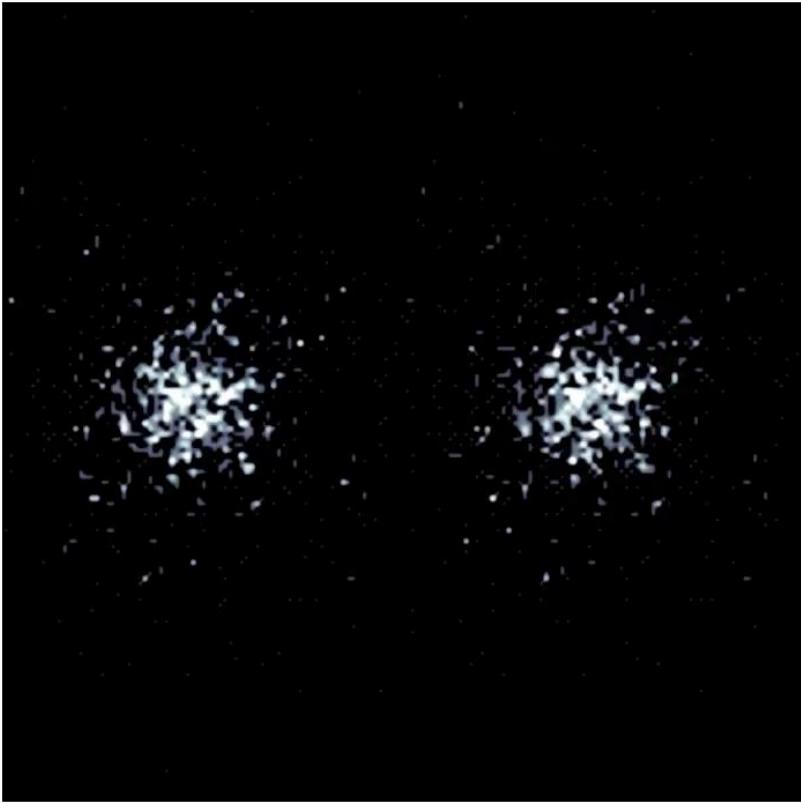
700 PB/year

HPC processing

DATA CAN'T BE SPLITTED

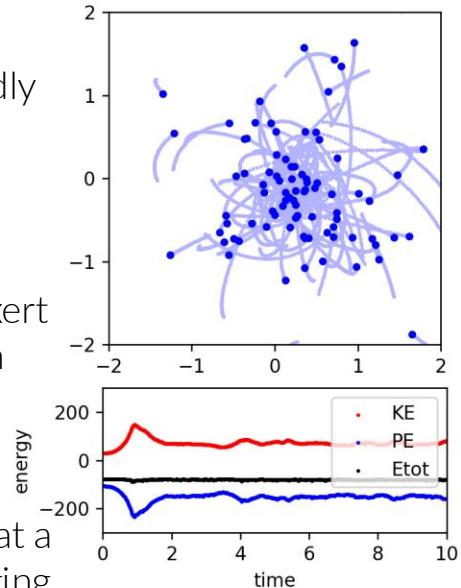
- **The computing challenge.** In order to improve the codes' performance, multi (multi-core CPUs) and many cores (GPUs) architectures have to be exploited.
- **The memory challenge -** Huge datasets cannot be loaded in the memory of a single CPU and cannot be handled by a single processor but by distributed memory systems. Distributed computing, based on the adoption of the MPI standard, represents a feasible and effective solution.
- **The data challenge -** This addresses the management, archiving and access of the raw data, the science data products, and the final outcomes of data processing and analysis.

HPC: the N-body problem



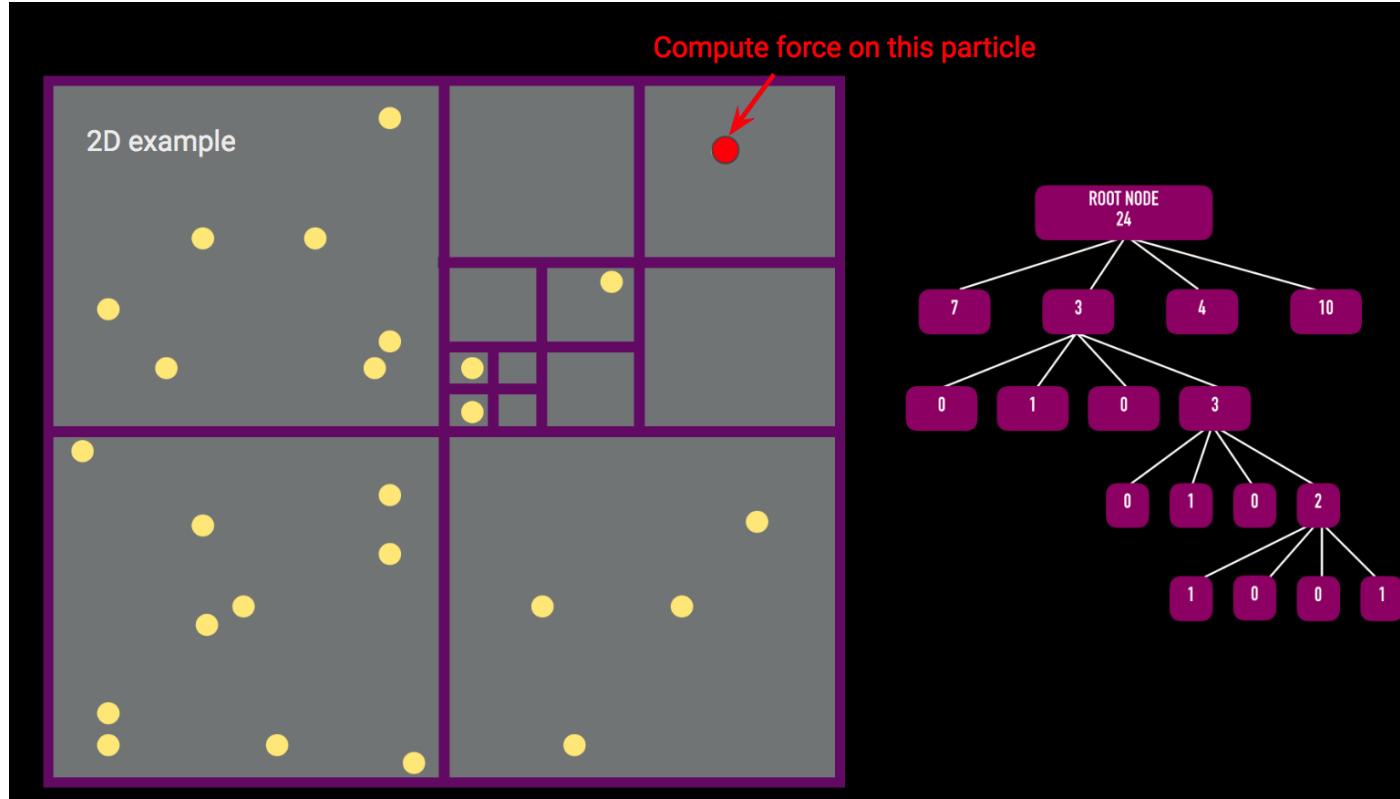
The N-body problem broadly describes the problem of predicting the future trajectories of a group of objects under the mutual gravitational forces they exert on one another, given each individual object's current position and velocity.

In Astronomy, the N-body problem has been studied at a wide variety of scales: ranging from the study of asteroids near Jupiter ([Brož et al. 2008](#)) to the study of the largest gravitationally bound clusters in the Universe ([Angulo et al. 2012](#)).

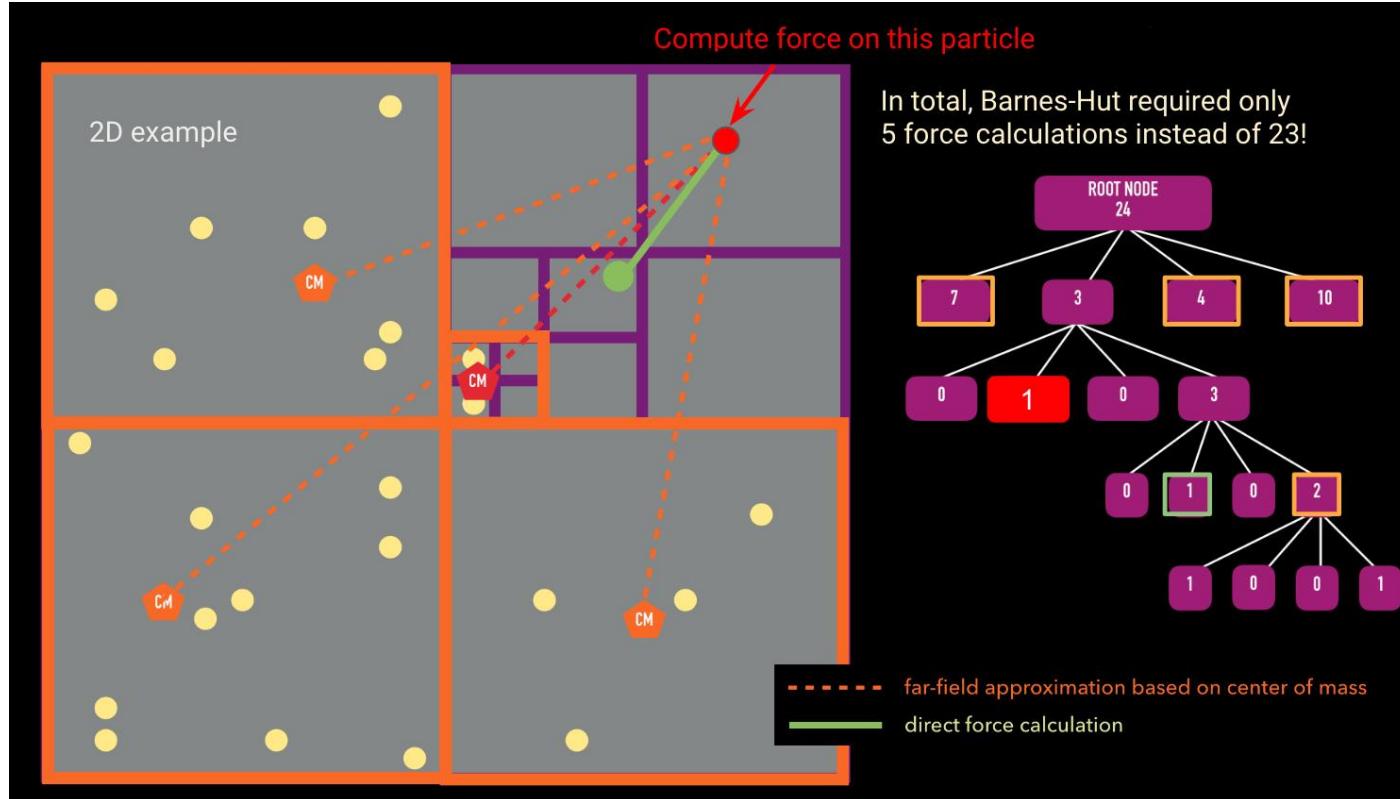


N-Body in practice

INSTEAD
OF
CALCULATING
THE FORCE
OF EACH
PARTICLE,
IT COMPUTE
A TREE ZONE
REDUCING
FROM N^2
to $N \log N$



N-Body in practice



| Parallel implementation

Including the communication overheads, a simple parallel algorithm has the following steps:

1. Build the Octree (on primary MPI node)
 1. Octree is broadcast to all MPI nodes
2. For each particle: compute the total force by traversing the Octree
 1. Particle positions and velocities are distributed across MPI nodes
3. Update the velocities and positions using LeapFrog time integration algorithm (on the GPU)
 1. Particle positions, velocities, and accelerations are computed
4. Entire particle catalogs are sent back to the primary MPI node

Tight coupled: at each time step nodes must communicate!

THE IDEA : instead of each node
there's a cluster with
a batch and a master
batch that manage all

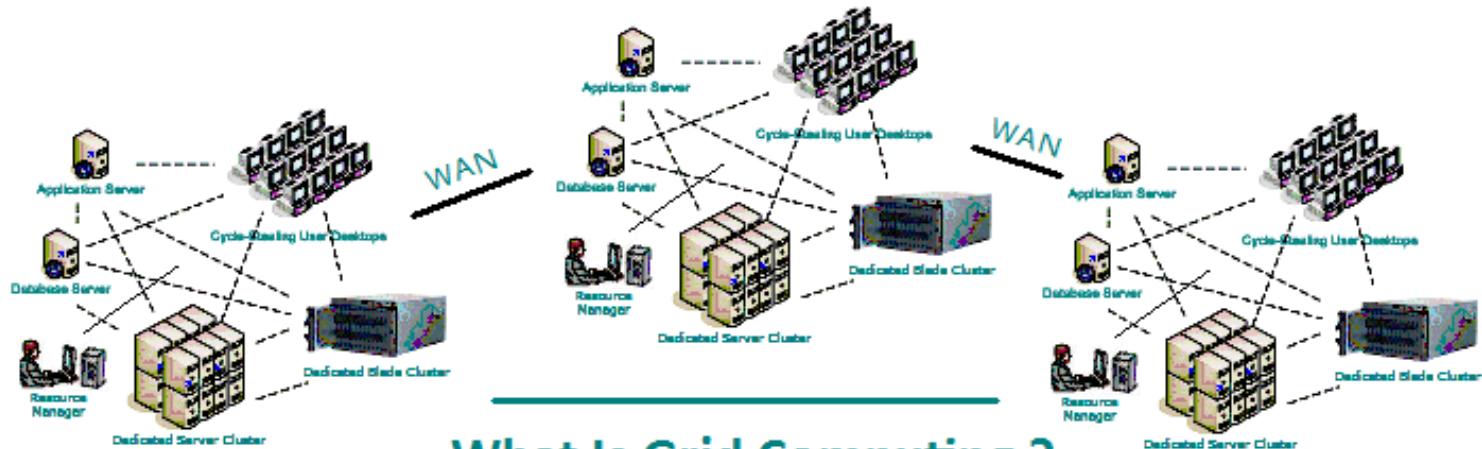
GRID Computing

"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."

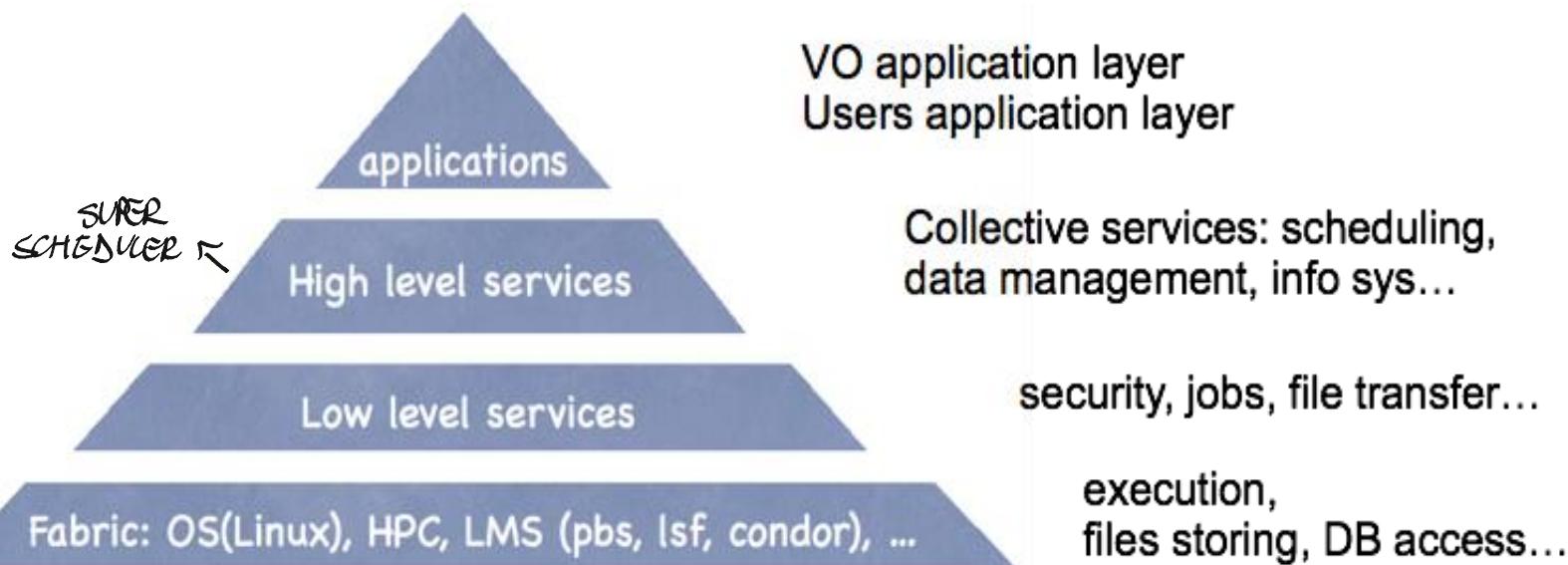
"[framework for] flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources"

"collection of geographically separated resources (**CLUSTERS**) connected by a network [...distinguished by...] a software layer, often called middleware, which transforms a collection of independent resources into a single, coherent, virtual machine"



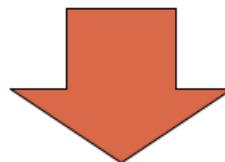
What Is Grid Computing ?

It's the software layer that glue all the resources
Everything that lies between the OS and the application



Virtual Organization

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



VIRTUAL ORGANIZATIONS

GRID MIDDLEWARE

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

EMULATIONS AND VM WERE WHAT KILLED THIS CONCEPT

“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

| Utility Computing: concepts

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: concepts

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.

Different model may be implemented even if the pay per use is the most common one (e.g. flat rate, metered, etc)

The pricing model is what characterize the Utility Computing

Utility Computing

Data backup

Data Security

Partners competences

Defining Service Level Agreement

Getting value from charge back