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)

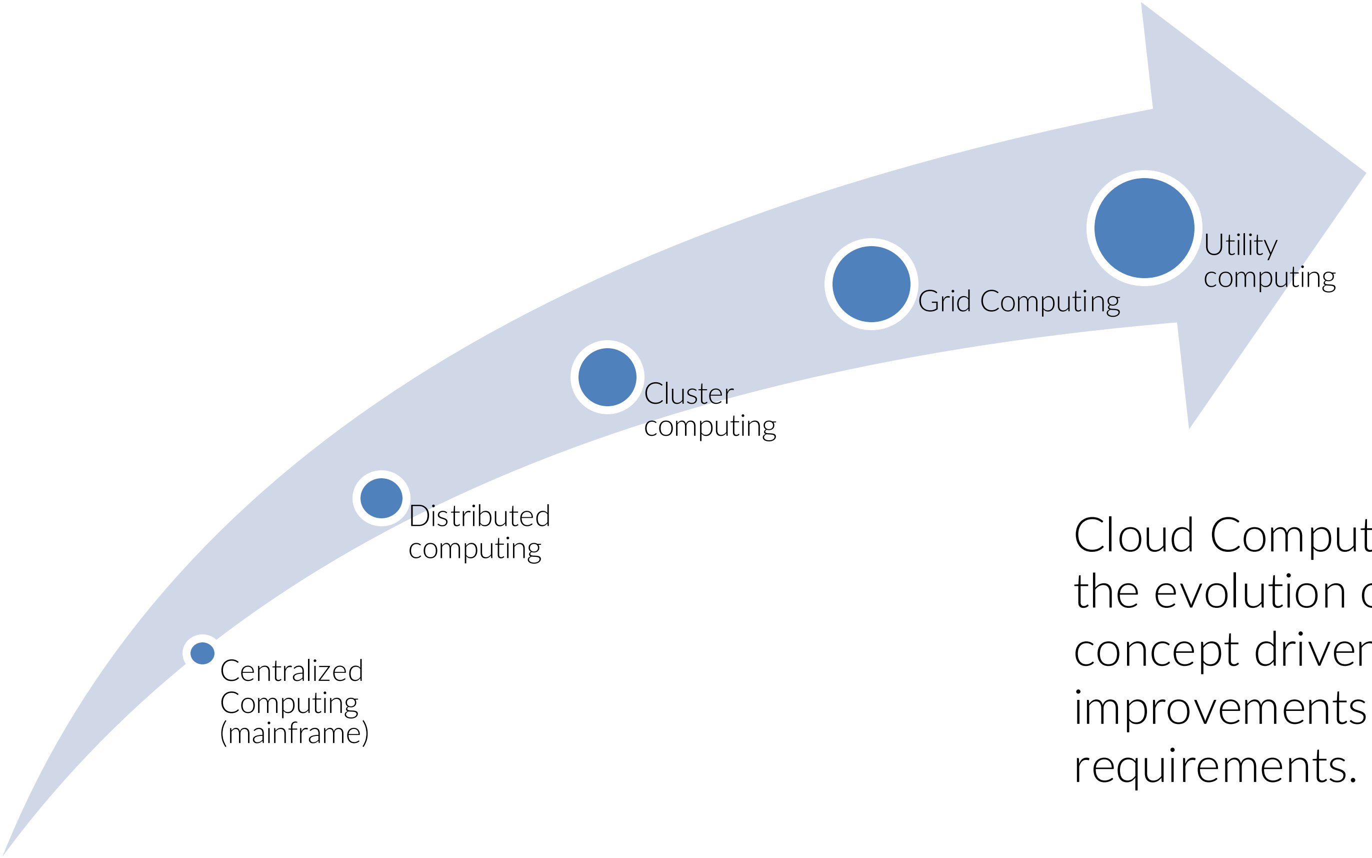
Modern computing

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



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

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)

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

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

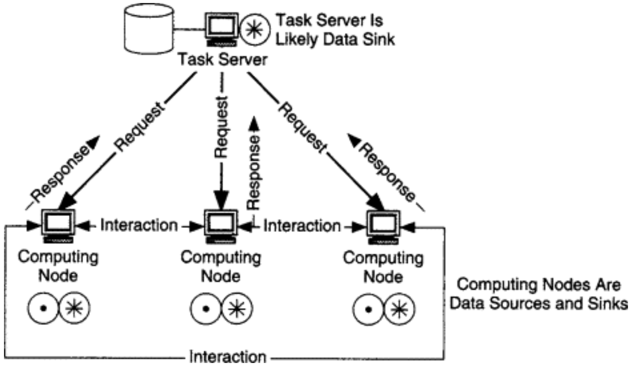
Load Sharing and balance - to distribute computing on different nodes to share loading to the whole system

Scalability - Easy to expand

Performance - More resources involved

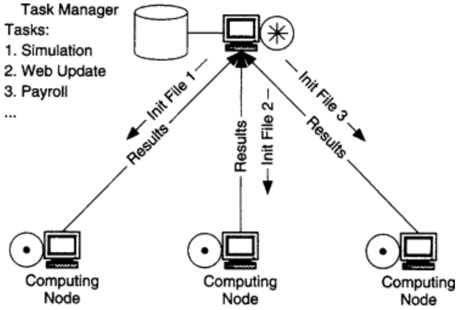
| DD Architecture

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



client-server: resource management centralized at a server

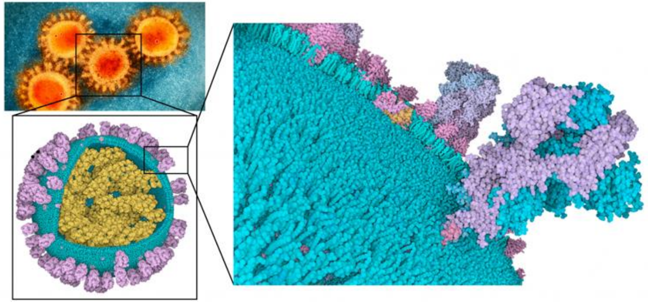
3-Tier architecture: move the client intelligence to a middle tier to simplify application development.



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

Distributed applications

OpenPandemics - Covid19



High Availability Systems
Distributed databases
High Throughput Computing

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

Apache hadoop

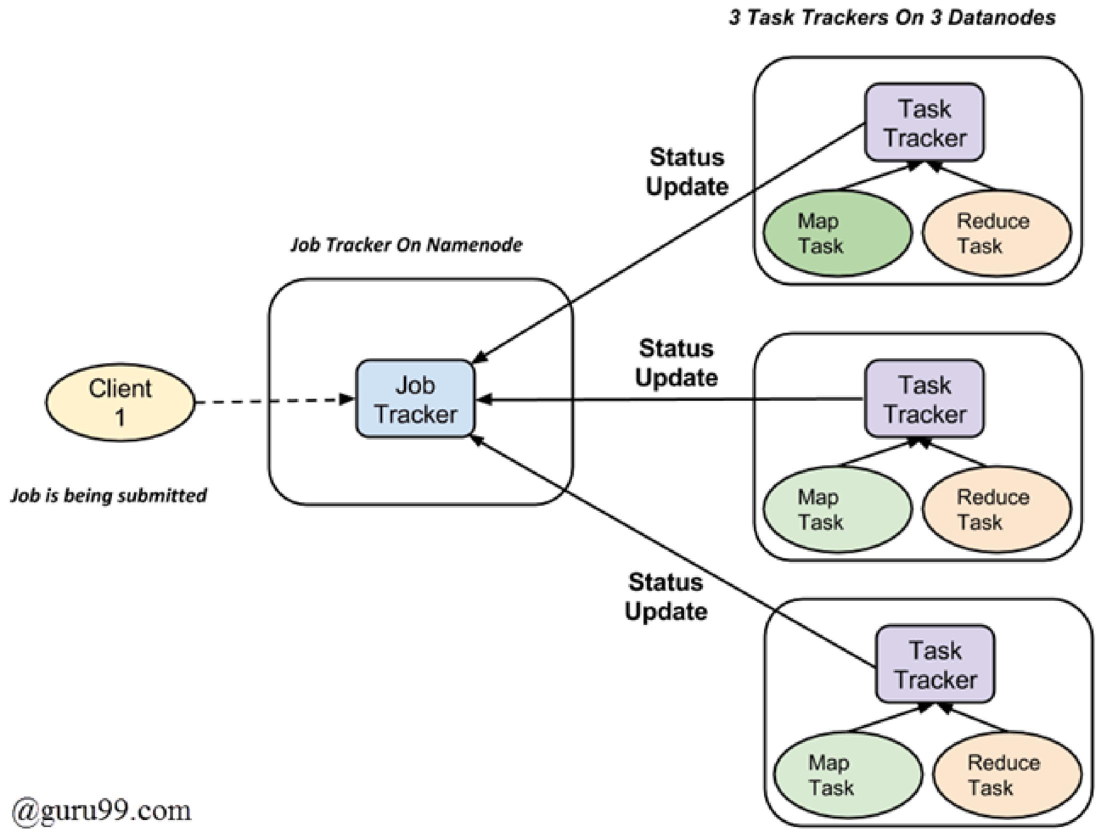
Apache Hadoop system implements a distributed scalable computing model for data analytics

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 work



multiple tasks onto multiple data nodes

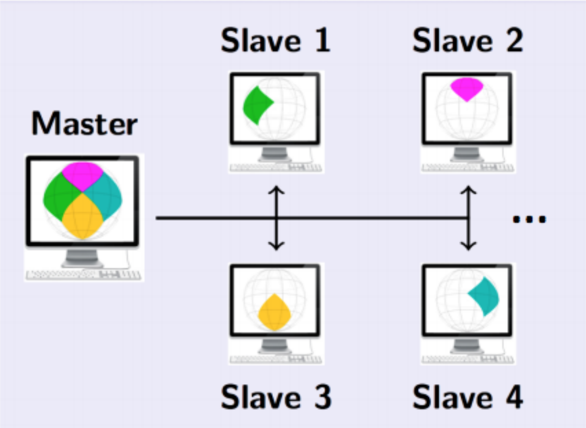
job tracker - coordinate and scheduling

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.

Data distribution and processing

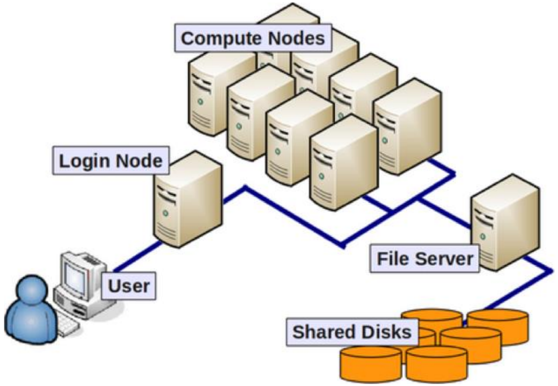
- With Hadoop / Spark, the data is distributed over several nodes
- Distribution ?
- How to optimise it ?



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

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.

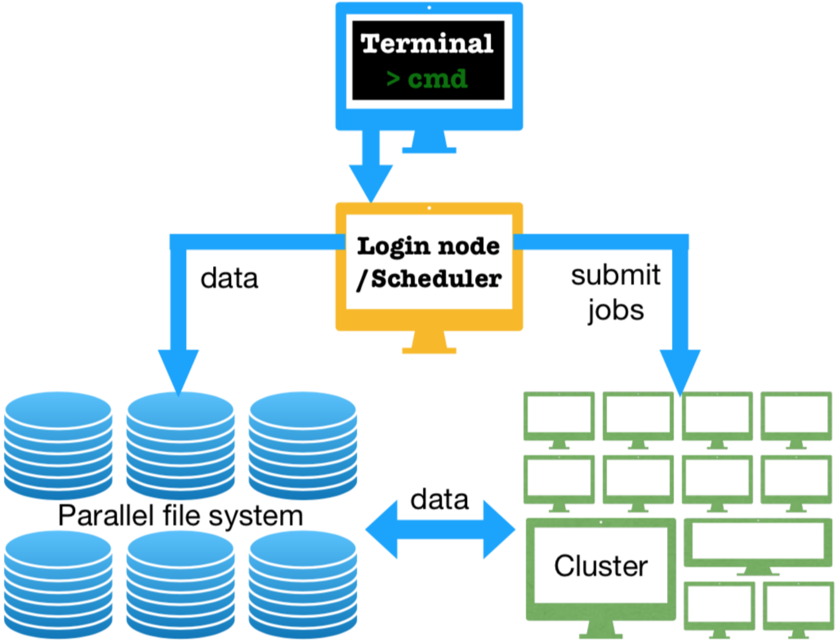
Network Load balancing clusters

- 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

Using a Cluster



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

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 vs HTC

High Performance (Capability)

High Throughput (Capacity)

Fine-grained Applications
- **Many-node**
- **Few concurrent runs**
- **High interconnect use**

Course-grained Applications
- **Single-node**
- **Many concurrent runs**
- **No interconnect use**

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);
- combine the partial results produced by the system to give the overall result.

Can I partion my data?

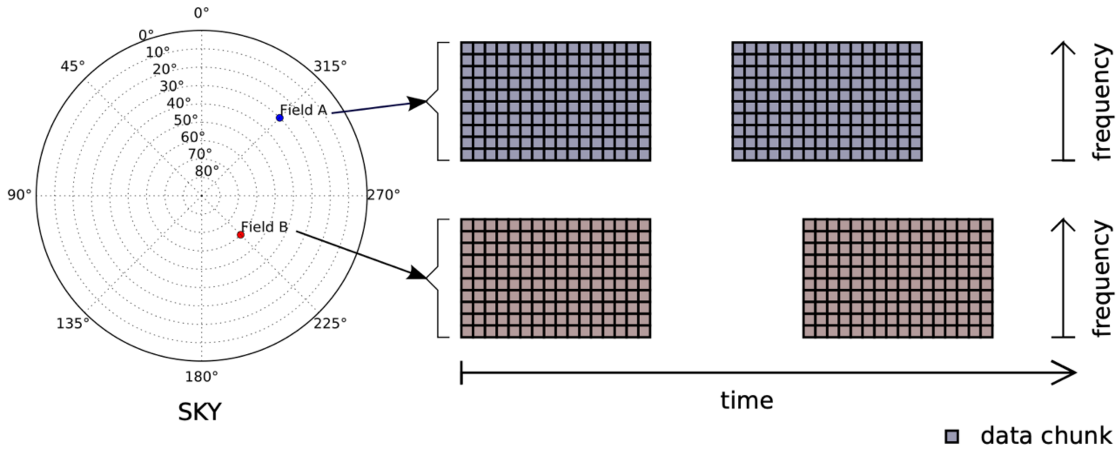
Are the small parts independent?

Do you estimate the overhead of partition the data?

LOFAR Pipeline parallelism

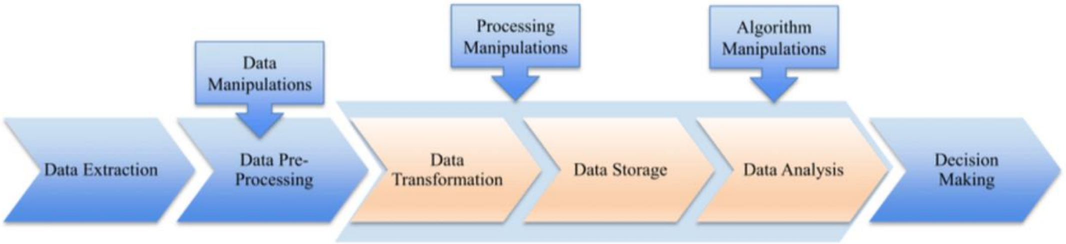
Each 8 hours observation is 115 MHz to 175 MHz in 371 separate sub-bands. Each sub-band was originally composed of 64 spectral channels, and the initial scan-time was set to 1 second. We need to process 320 observations.

SKY+TIME+FREQUENCY data parallelism

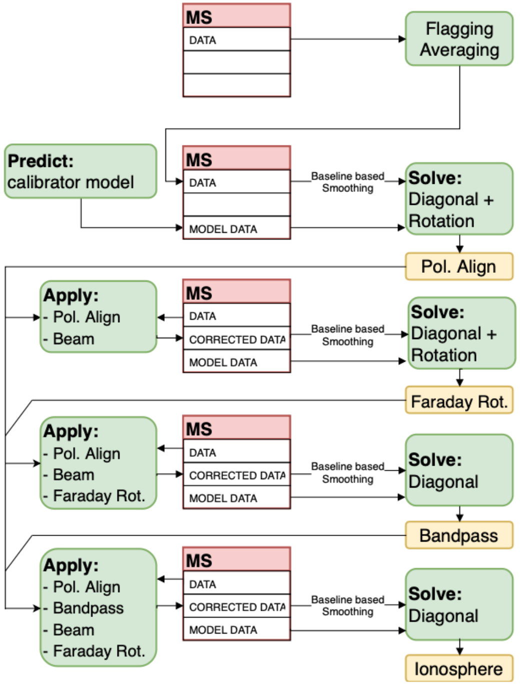


Data reduction PIPELINE

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?

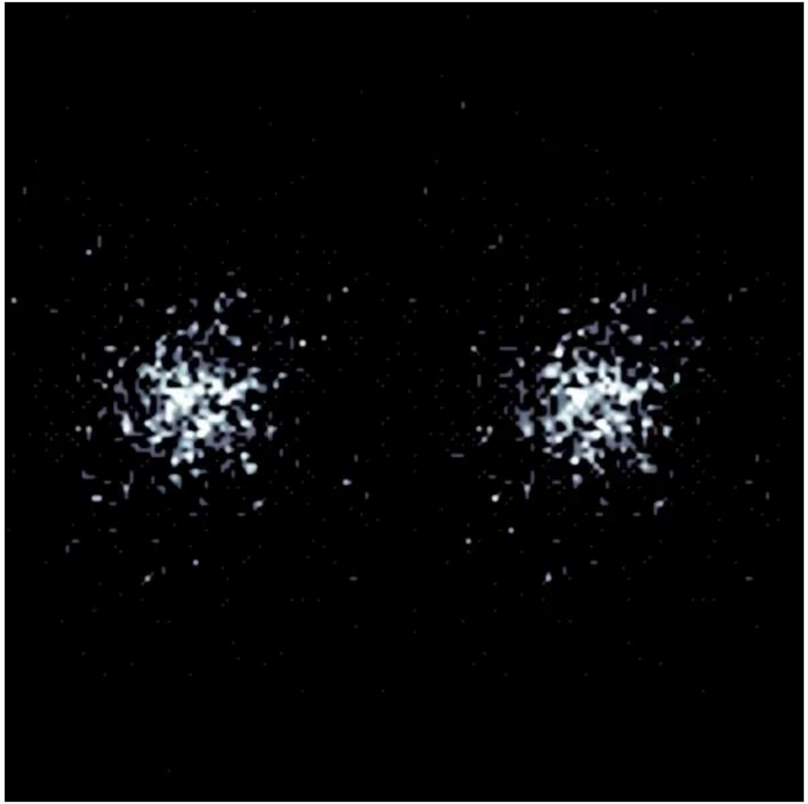
High Perfomance Data Analysis

- 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

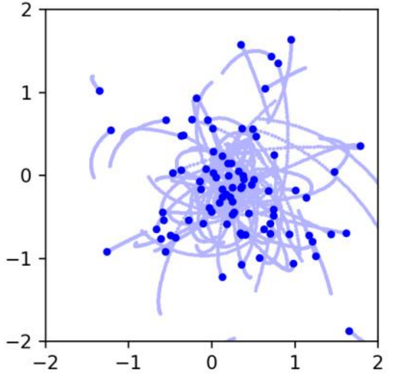
HPC processing

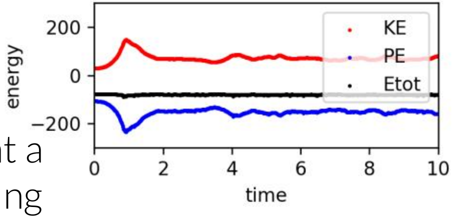
- **The computing challenge**. In order to to improve the codes' performance, multi (multi–core CPUs) and many cores (GPUs) architectures have to be exploited.
- **The memory challenge -** Huge datasets cannot be loaded in the memory of a single CPU and cannot be handled by a single processor but by 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).





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!

GRID Computing

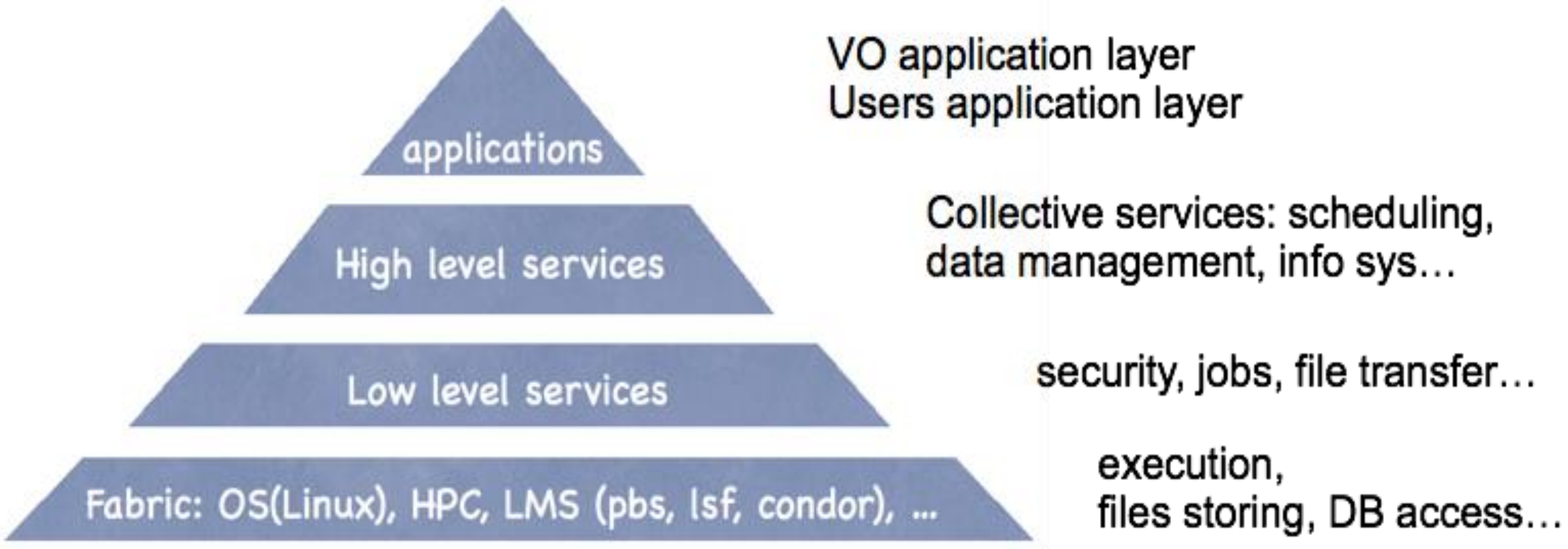
“a single seamless computational environment in which cycles, communication, and data are shared, and in which the workstation across the continent is no less than one down the hall"

“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”

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

“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

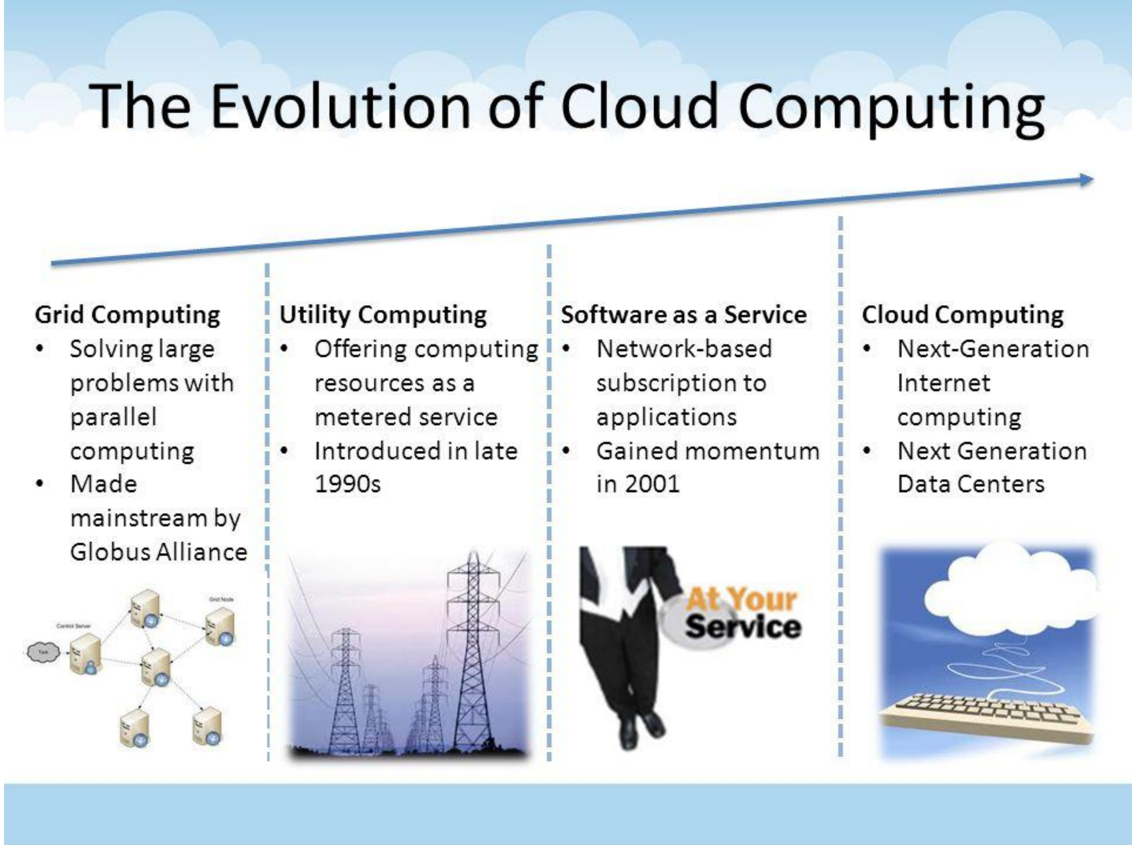
Data Security

Partners competences

Defining Service Level Agreement

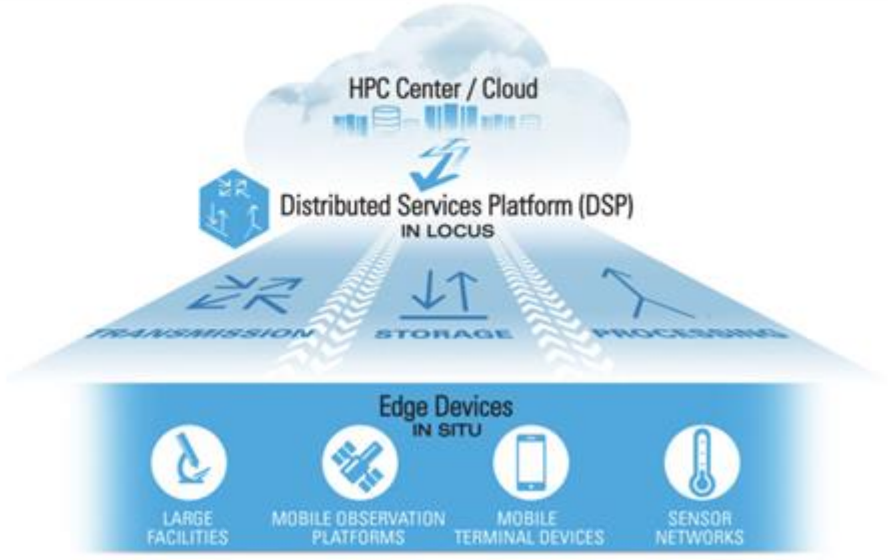
Getting value from charge back

Towards Cloud



Edge Computing

- Edge Computing is a **distributed computing paradigm** in which processing and computation are performed mainly on classified device nodes known as smart devices or edge devices as opposed to processed in a centralized cloud environment or data centers.
- It helps to provide server resources, data analysis, and artificial intelligence to data collection sources and cyber-physical sources like smart sensors and actuators.

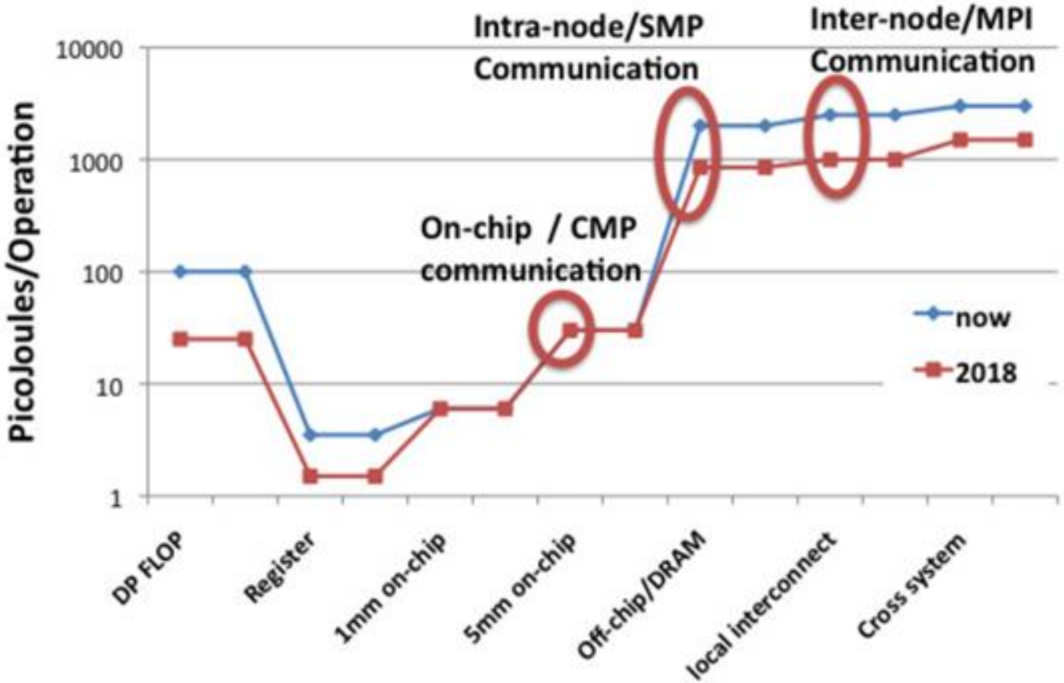


What is Edge Computing

- A network of **micro data centers embedded in the instruments/sensors** that store or process critical data locally and push received data to a centralized data center or repository of cloud storage.
- Edge computing processes the data locally results in reduced traffic in the central repository.
- Computing close to data: reduce data transfer!

Data Transfer has a cost

The closer it is, the less it "costs" (in terms of latency and power).



Edge Computing Features

- Computing at the edge of network
- It focuses bringing computing as close to the source of data
- It decentralizes processing power to ensure **real-time processing** without latency while reducing bandwidth and storage requirements on the network
- It processes data locally at local network infrastructure instead sending all the data to the cloud data centre
- A device which produces data is edge devices like machines and sensors, or any devices through which information is collected and delivered: EDGE DEVICEs