LABORATOIRE
D'OCEANOGRAPHIE
SPATIALE

Data intensive science platform

Benefits from Big-data and Cloud technologies to build a petascale data analysis platform

Cargo Day n°5 - Rennes - 19 Novembre 2015

Frédéric Paul (fpaul@ifremer.fr), Olivier Archer, Jean-Francois Piollé, Bertrand Chapron
Institut Français de Recherche pour l'Exploitation de la Mer (Ifremer), Brest, France

Why are we here ?

- Cargo Day, 5th edition

  “ Pour cette journée, nous allons rester très pragmatique et essayer de couvrir ce sujet vaste sur les données, mais tout en restant focalisé sur le comment notre métier et nos expériences du terrain ont su répondre et vont répondre demain à cette augmentation drastique des données de plus en plus massives que nous manipulons quotidiennement, tant sur le traitement que sur l'analyse ”

- This presentation focuses on technical architecture, design and feedbacks from our experimental platform dedicated to Data Intensive Science applications, built on Big Data and Cloud computing key concepts
  - why this project was launched
  - how cutting-edge technologies helped to cope with new challenges
  - how the platform evolved from a few dozen terabytes store to several petabytes analysis tool
  - how this platform changed users habits
  - what is the current status and following

Context : Laboratory of Oceanography from Space

IFREMER

- Institut Francais de Recherche pour l' Exploitation de la Mer
- Ocean resources and exploitation knowledge and discovery : research and expertise
- Marine and coastal environment monitoring, sustain development of marine activities

- 1530 employee (+ 330 from Genavir)
- 5 centers in France, 26 coastal sites
- Fleet : 8 vessels, 4 submersibles (Nautile, Victor 6000, AUV)

LOS & CERSAT (Brest)

- Laboratory of Oceanography from Space
  - Scientist, engineers, technical teams
  - Expertise in sea surface physics and air-sea interactions
- CERSAT is the satellite data center of Ifremer, addressing international user community
  - Focus on sea surface parameters : wind, waves, fluxes, sea surface temperature, sea ice, salinity,...
  - More than 30 satellite missions & 300 data collections
- Some partners :

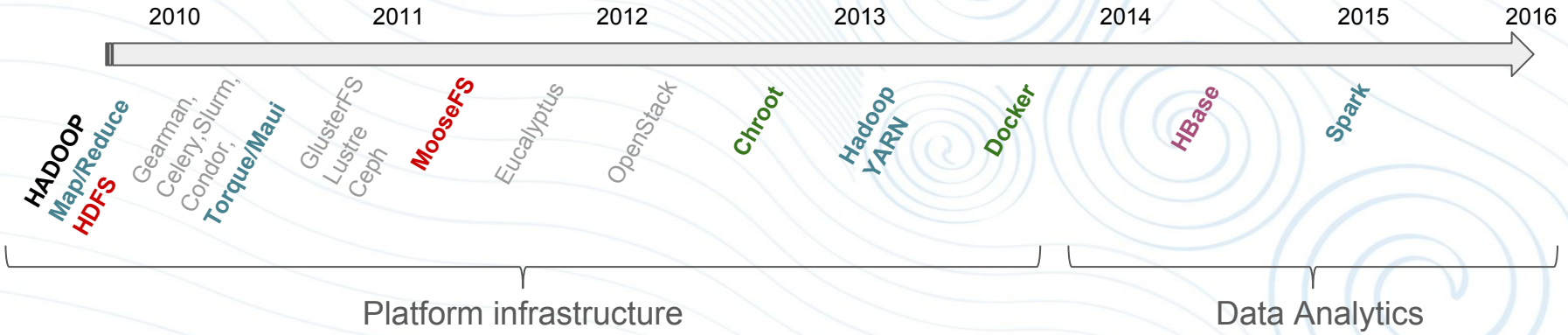
Context : Laboratory of Oceanography from Space

- About Data
  - ~30 satellite missions & ~300 data collections + Model, in-situ..
  - ~25 years of ocean data
  - data volume : 400TB - 1PB
  - heterogeneous contexts : operational / near-real-time, bulk reprocessings, Science
  - input streams : dropper daily downloads (eg. 100GB) & nas media (eg. 100TB)
  - scientists, exploitation team, Ifremer users & external partners and projects
  - tools & Services (Indexation, Visualization, Analytics, Data mining, Processing chains, ...)

- BigData & Cloud Plaform project
  - **one platform to handle most of our data activities (ideally)**
  - started in 2009-2010 as experimental platform
  - all data archives stored on 1 platform
  - usable by all projects and users (local & remote), each using dedicated virtual environments
  - high Performances for daily research activities as well as massive bulk processings
  - easy management, fault tolerant, scalability
  - reasonable cost, yearly upgrades

Challenges and technologies overview

1. Petascale data archives
2. Effortless distributed processing
3. Share with remote partners
4. Batch Big-Data analytics
5. Interactive data insight



PLATFORM SOFTWARE STACK

Datacrunch & Datavore

Gogolist | Celery | HBase | Hadoop | Pig, Spark

Torque/Maui (HPC, MPI ... (for Hadoop ecosystem)

Bare-metal | Chro ... ~~OpenStack KVM~~

MooseFS (ce ... Hadoop HDFS

Ubuntu (12.04) | Ubuntu (12.04) | Ubuntu (12.04) | Ubuntu (12.04)

Server 1 | Server 2 | ... | Server 150

[2010] Dell R510, 12 cores, 24GB ram, 10TB raid5

[2014] Dell R720xd, 16 cores, 128GB ram, 72TB JBOD

App. Frameworks

Ressource Managers

Virtual Environment

Distributed Filesystem

Hardware / OS (moderately heterogeneous)

Platform design : Big Data framework key concepts

Hadoop = Ecosystem !

Architecture Key Points :

- Commodity hardware
- Horizontal scalability
- Usual network (Gbps)
- Easy Maintenance
- Cost effective

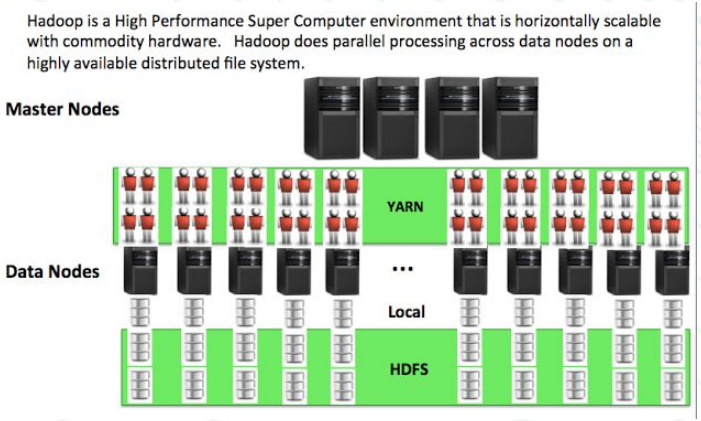
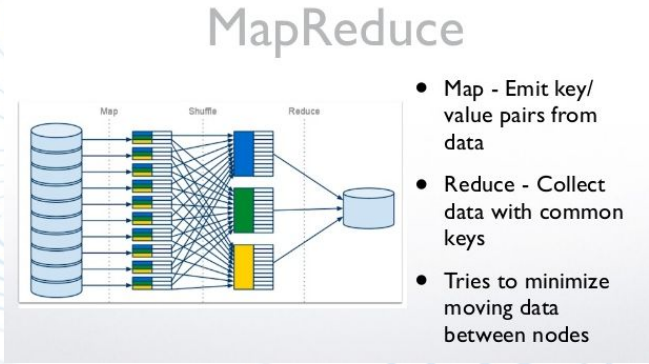


Image from : http://cloud-dba-journey.blogspot.fr/

Processing model Key Points

- Map/Reduce programming model
- Move code to data
- Smart scheduler
- Resilient, natively fault tolerant
- Data locality benefits

New approach compared to HPC



Platform design : Cloud computing key concepts

Infrastructure as a Service, Virtual Machines, Virtual environments… Elasticity !

- How IAAS can be useful for satellite data processing ?
  - customized Processing Cluster on demand
- Experiments : Eucalyptus (05/2011), OpenStack (04/2012)
  - Many issues : Maturity, Reliability, Complexity, Performances,…
  - Sexy idea, but implementation complicated… Alternatives ?
- Replace virtual machines by virtual environments
  - should solve many issues ! Network, performances, reliability, resources management…

- In practice, it all depends on the REAL requirements
  - Virtual machine on demand : OpenStack
  - **Container isolation : Docker** -like solution (based on lxc)
  - **Virtual processing environment : Chroot**
    (most of our use cases finally)

GNU/CHROOT

Note : the chroot system call was introduced during development of Version 7 Unix in 1979, and added to BSD by Bill Joy on 18 March 1982 – 17 months before 4.2BSD was released – in order to test its installation and build system.

Platform design : Big Data, Cloud, HPC…

Big Data	Cloud	HPC
Makes easier to exploit huge volumes of data (in our case)	On-demand adaptative environments	Processing Performances
Key concepts (Hadoop) : Architecture : **commodity/cheap hardware, horizontal scalability** Storage : **distributed, scalable, fault-tolerant, low-cost** Processing : **smart schedulers, data locality, map/reduce**	Key concepts (eg. Amazon) : "*-As-A-Service" Architecture and Virtual **Environments on-demand**, **Elasticity**, Scalable Online Cloud Storage "Low cost", pay per use	Key concept : The best of the CPU The best of the Network The best of the Storage Optimize all at all levels
Maturity … Carryall buzzword	Cost (long term) New engineering technical efforts	Cost, optimization effort Technical complexity, old school

IDEA : COMPLEMENTARITY (one does not replace the other)

Challenge 1 : Petascale data archives

Goal : from narrow NAS & inefficient LTA to powerful, cheap and scalable storage

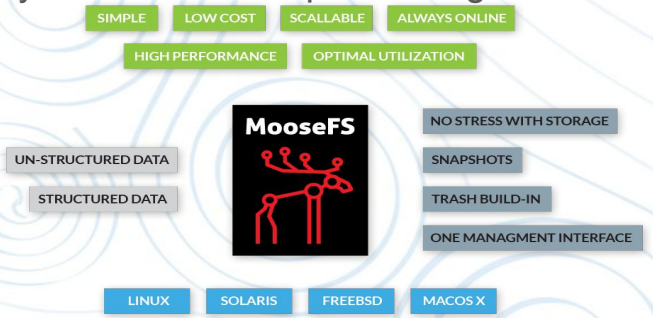
- Open-source
- Horizontal scalability
  storage capacity, performances
- One unique filesystem
  (NFS-like mount for users)
- Data replication : fault tolerance, safety, performances, ease of management
- Cost friendly (jbod, no raid)

Issues :

- Finding a suitable DFS is not easy… for our needs at least : EO satellite data, existing practices
- HDFS, Lustre, GlusterFS, Ceph, Isilon, ... not suitable in our case (incompatible with existing practices, high latency, complexity, no reliability, proprietary/prices)

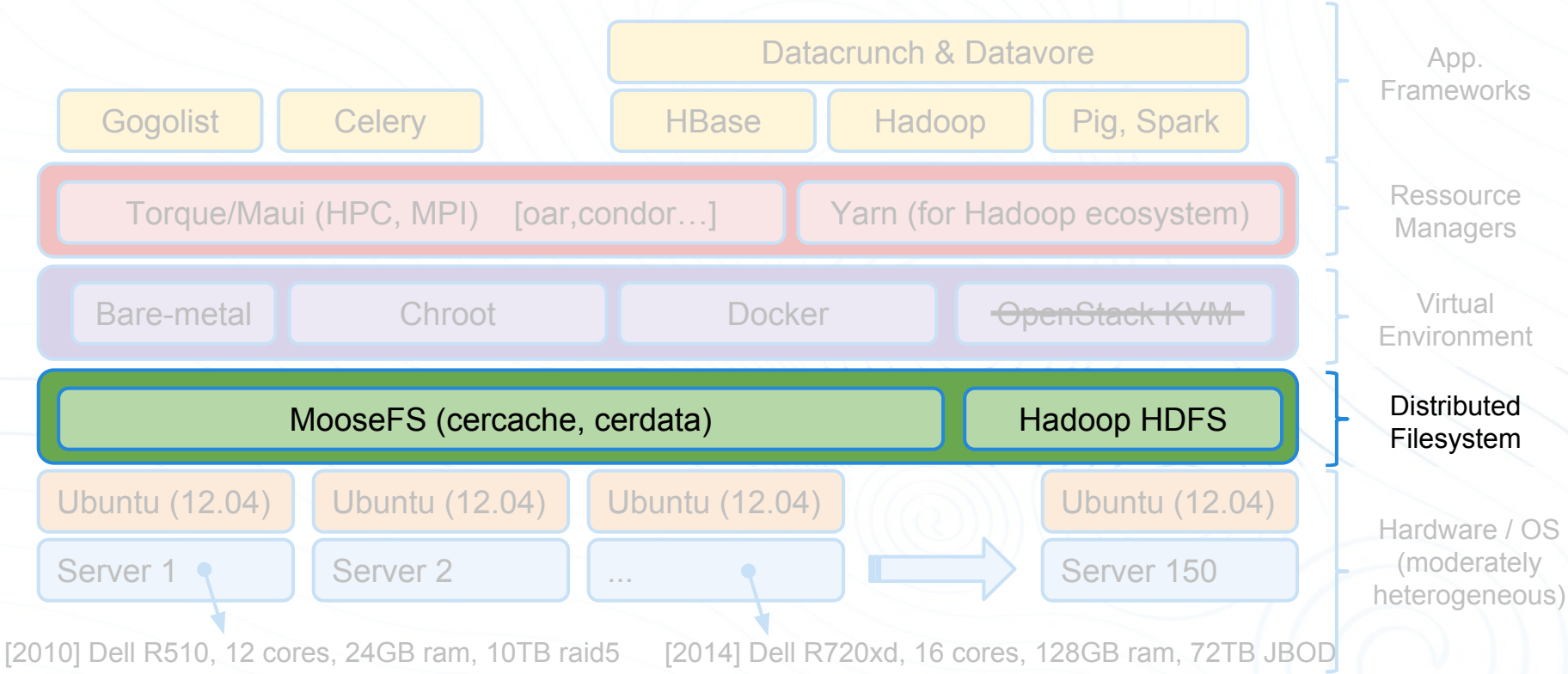
MooseFS

- Our final choice (03/2011), still running
  Why ? No crash after weeks of (heavy) use ...
- Smooth scalability from TB to PB (effortless)
- Easy management
- Reliability (resilience, data health checks, ...)
- Yet not perfect… : performances, replication vs distributed-raid, no standard NFS mount, ~posix
- Usage : online data archive (including LTA mirror), everyday's work, mass reprocessing



Note : Hadoop HDFS still used, but for internal chain processing purposes, not for data archive storage

PLATFORM SOFTWARE STACK



Challenge 2 : Distributed processing made easy

Job schedulers

Goal : efficient processings, easy distribution of independant jobs accross nodes, for scientists and bulk processings

Issues :

- Hadoop Scheduler immature (2011..)
- HPC Schedulers too complicated/heavy (slurm, condor, torque/maui)
- Light solutions (eg. Celery, Gearman) : lack of ressources management...

Implemented solution :

- HPC Scheduler Torque/Maui with “Gogolist” homemade wrapper was really effective for users !
- Apache Hadoop Yarn finally arrived (too late, still underused c/w Gogolist)

On-demand dedicated environments

Goal : no more painful software integration using virtual environments, on-demand instances

Concepts : Cloud IAAS, Virtual Machine & Container

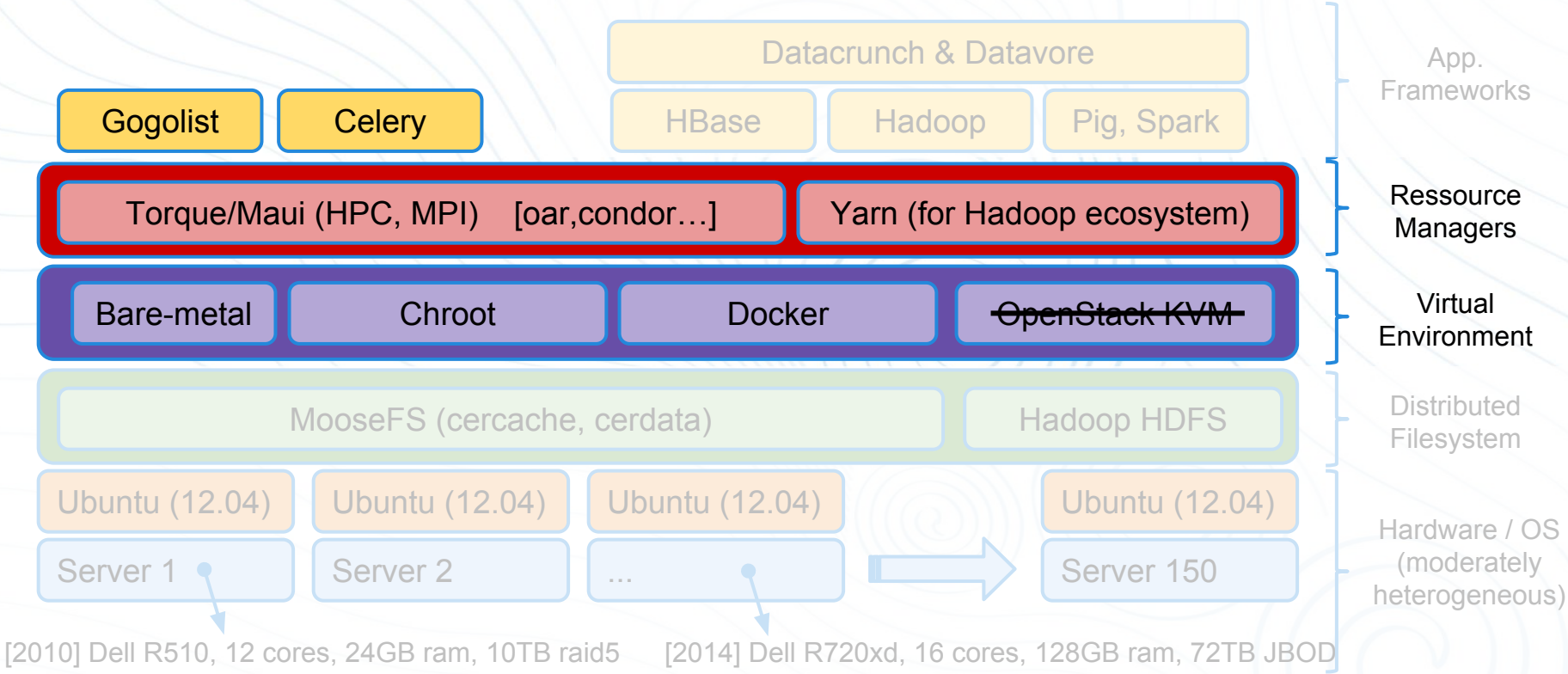
Issues : OpenStack immature (2012) & not suitable

Implemented solution :

- GNU/Chroot
- Docker (2013)

“Killer-feature” : allowed to easily perform massive bulk processings using legacy ESA processors… helped to develop platform usage through many use cases (eg. ERS1/2, Landsat, Envisat processings)

PLATFORM SOFTWARE STACK



INFRASTRUCTURE LESSONS

- Organization
  - building the platform can be done internally (is best?), based on use cases knowledge
  - operational exploitation requires additionnal dedicated ressources (our case : ¼ human.year)
- Practices
  - Stay really close to users : deep understanding of needs is critical
  - Best practice : re-assess the best pratices… be pragmatic !
  - Keep things as simple as possible, don't expect universal solution…
  - Continuous improvement
- System Administration
  - Monitoring is critical (Nagios, Munin, Ganglia, Homemade...)
  - Automatic provisionning (Cobbler)
  - Automatisation (no Puppet/Chef&Cie for us, ansible and cssh are enough for small team)
- Hardware Investments
  - Negociate a partnership the very first day. Research prices can be really discount !
  - Increase capacity over time, not in advance (eg. lower disk prices)

Challenge 3 : Share with remote partners

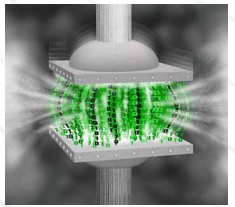
- Goal : share data archives, softwares, processing tools and documentations with remote partners to improve productivity and smooth exchanges
- Issues
  - Network security rules : major obstacle to such practices (intranet access…)
  - Working remotely can be painful (console access, latency…)
  - Public hosting/cloud is expensive (using lots of data) and can be unhandy
- Solutions
  - Hybrid platform (mix public & private cloud platforms)
  - Brand new datacenter in Brest (Pebsco) : host our hardware (big volumes at low cost) and provide direct internet links. Not without drawbacks : funding, system administration, speed…
  - Remote desktop solutions are effective : NoMachine/NxClient, X2Go
- Use cases examples
  - Partners & scientists to perform fast (massive) data analysis using available online data
  - Online collaborative data studies using Python, JupyterHub, ...
  - Web application prototyping without DMZ headaches, accessible huge data archives

Challenge 4 : Batch Big-Data Analytics

- Goal : leverage new ways to easily extract value from data archives, enjoying the benefits from the underlying BigData & Cloud platform
- Prototype example : DataCrunch framework

More on this during Lightning talk !

Using Hadoop Map/Reduce to efficiently crunch GB to TB satellite datasets to produce useful analytics for scientists within minutes (using Python,R,Matlab)

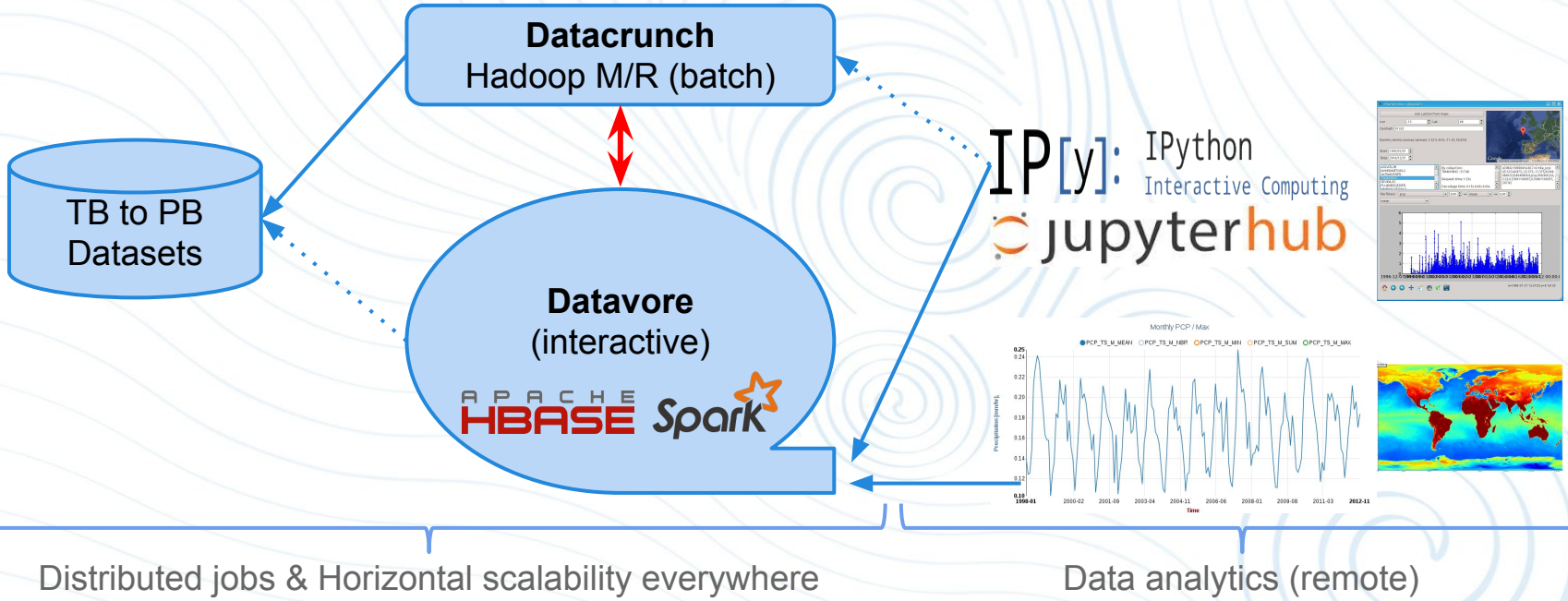


PIG

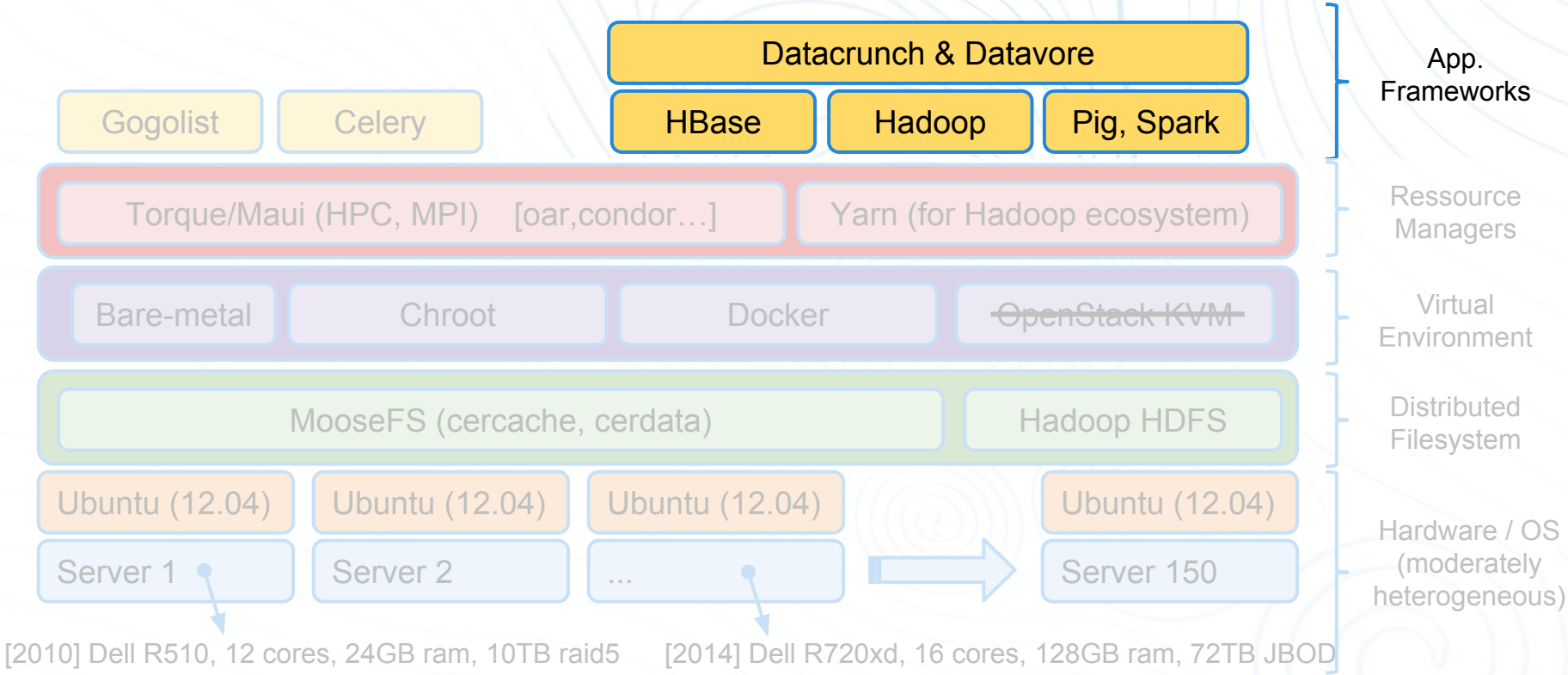
Challenge 5 : Interactive data insight

- Goal : provide interactive ways to analyse sparse data, using lightning-fast data warehouses coupled with batch-processing
- Prototype example : Datavore platform

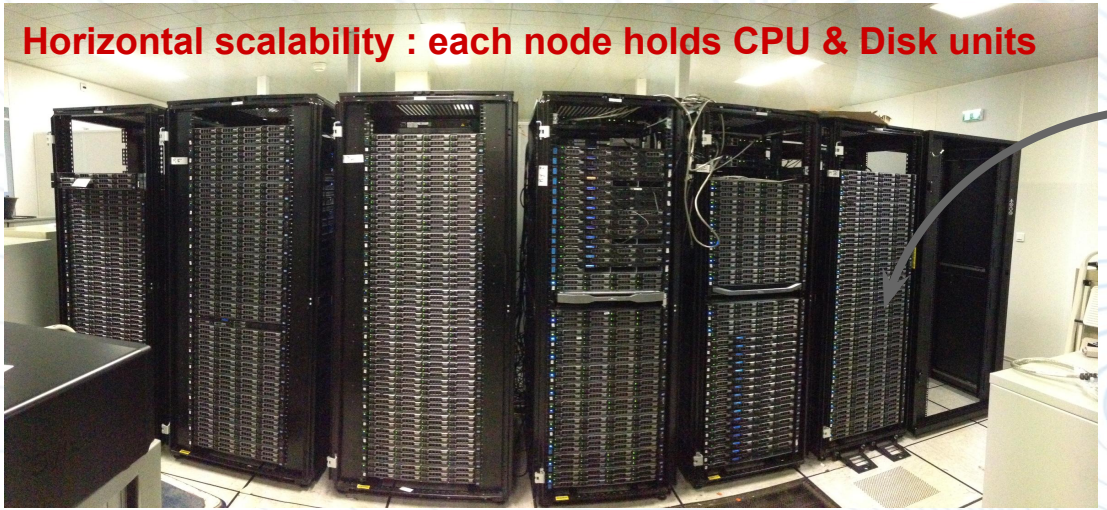
More on this during Lightning talk !



PLATFORM SOFTWARE STACK



PLATFORM PHYSICAL OVERVIEW



x150

Distributed Data Storage :

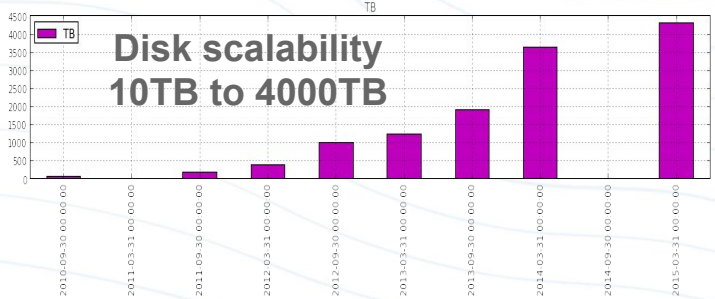
> 4 Po (~1300 sata drives)

Distributed Processing

> 1400 physical cores

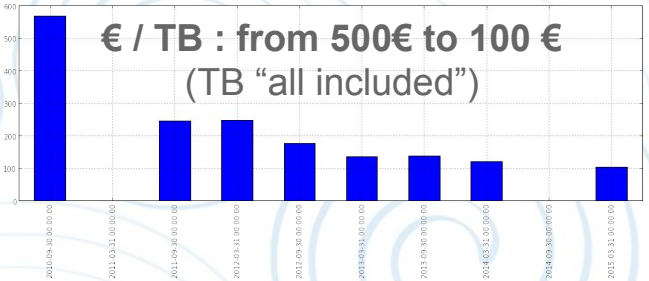
> 6TB memory

Platform based on **~150 nodes**



Hardware Cost : 700k€
over 5 years

Periodic upgrades :
2-3 times per year



CONCLUSION

- New data usages and increasing volume involved new technical approaches
- Required to face challenges at every level of the Platform Stack to build a suitable data analysis platform, relying on Big Data & Cloud concepts
- Not a multi-purpose solution, important security requirements still aside
- Lots of lessons learned, useful feedbacks, new natural ways to handle data
- Are users happy with this platform ? Goals achieved ?
- Benefits from these new opportunities are visible
- Future Platform Design is already work in progress, no major architecture change (lesser technical gap, Mesosphere, software-defined-data-center, ...)
- Focus on Data-exploitation-system and Data-as-a-service

QUESTIONS