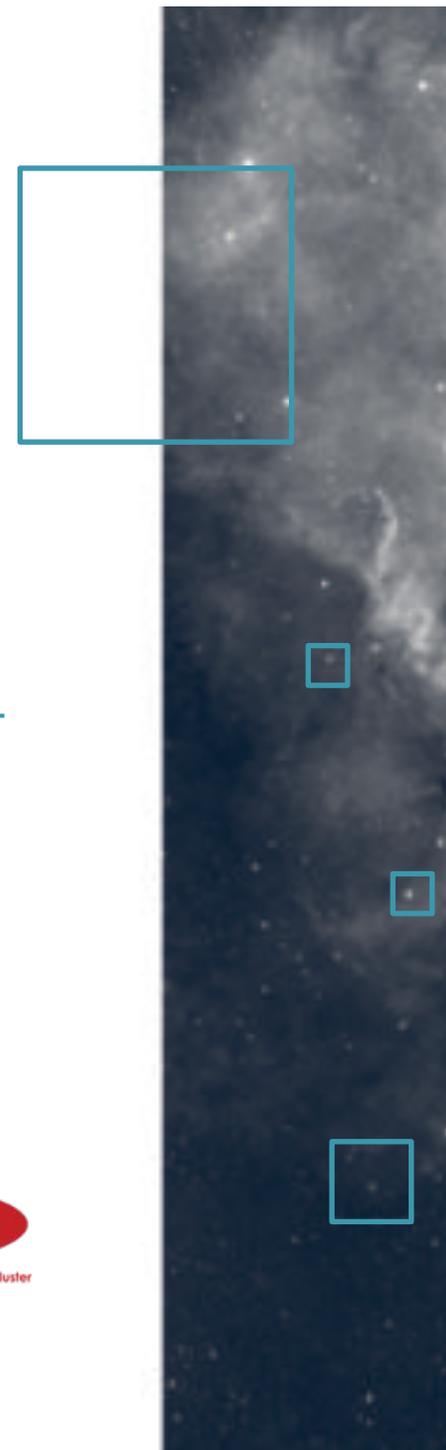


Experiments with Hadoop and Spark, discussion

André Schaaff, François-Xavier Pineau,
Noémie Wali

IVOA meeting, Sydney 2015



□ Around Big Data...

An ongoing exploration (started last month) of emerging (or maturing) « Big Data » technologies with the XMatch as main use case

- Emerging / maturing technologies



Credits: Apache foundation

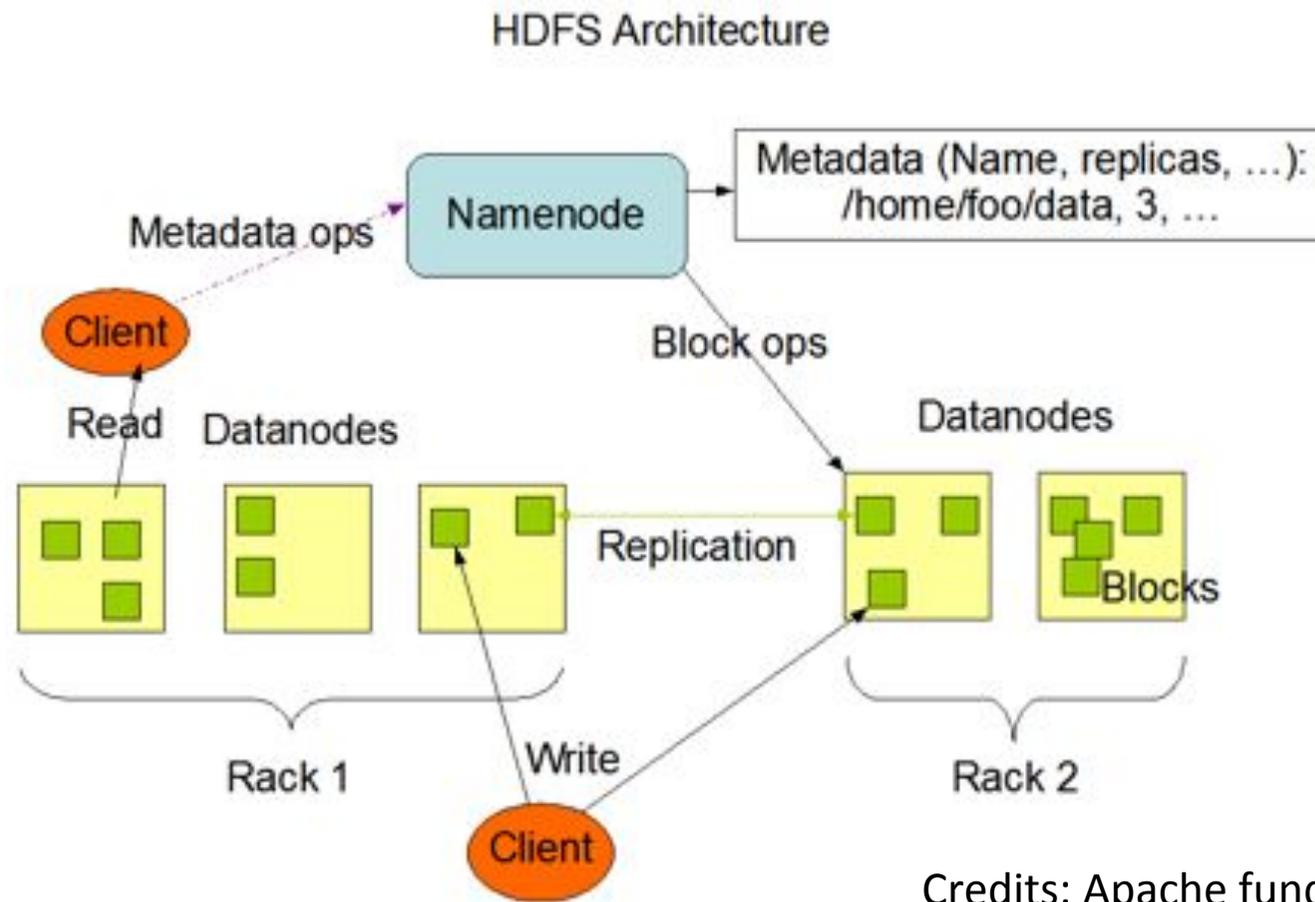
□ Hadoop ?

- **High-availability distributed object-oriented platform**
 - Framework for the distributed processing of **large datasets** (**HDFS** (distributed file system), MapReduce)
 - **Scalable** from a single server to thousands of machines

□ HDFS: more details

- **Abstraction of the storage**
 - A set of distributed harddisks is seen as **one harddisk**
 - **NameNode**
 - namespace, file tree, metadata
 - location of the data blocks
 - **DataNodes**
 - where the data blocks are
 - the DataNodes inform the NameNode of their content (data blocks)

□ HDFS Architecture



Credits: Apache foundation

□ Spark ?

- « Fast and general **engine for data processing** »
- Runs on **Hadoop (HDFS)**, Mesos, standalone or in the cloud
 - => Compatible with Hadoop data
- Applications can be written in Java, Scala, Python or R
- **Not only MapReduce** driven

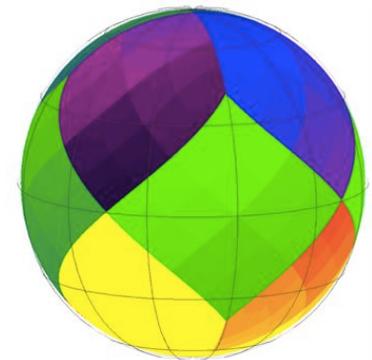
□ CDS XMatch

- The CDS XMatch is an efficient service based on **optimized developments** and implemented on a **well shaped hardware**

- Data size
 - do not fit into memory
- Performance issues
 - data loading
 - looking for candidates



- Scalability: Healpix partitioning
- Efficiency:
 - special indexed binary file
 - *kd*-tree (cone search queries)
 - multithreading



□ CDS XMatch (2)

- The data is **not distributed** on a few servers: all the catalogues are located on one RAID system

Single machine

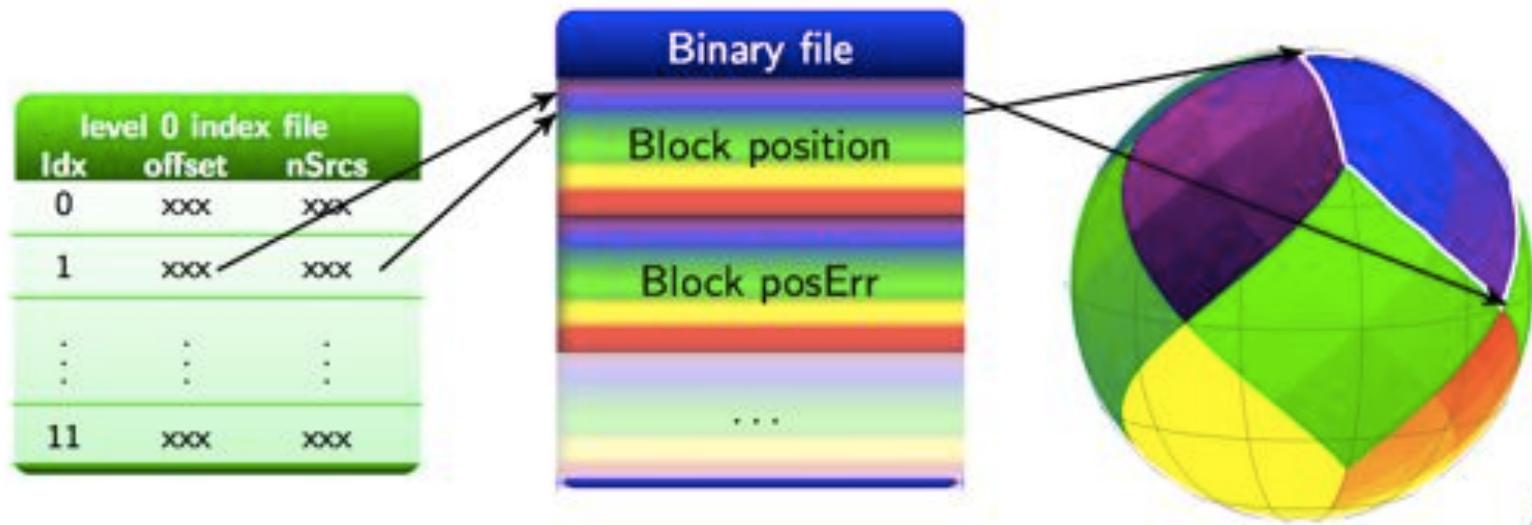
- All sky correlation (small catalogues)
 - allow “on the fly” correlation
- Correlation pixel by pixel (large catalogues)



- But ...

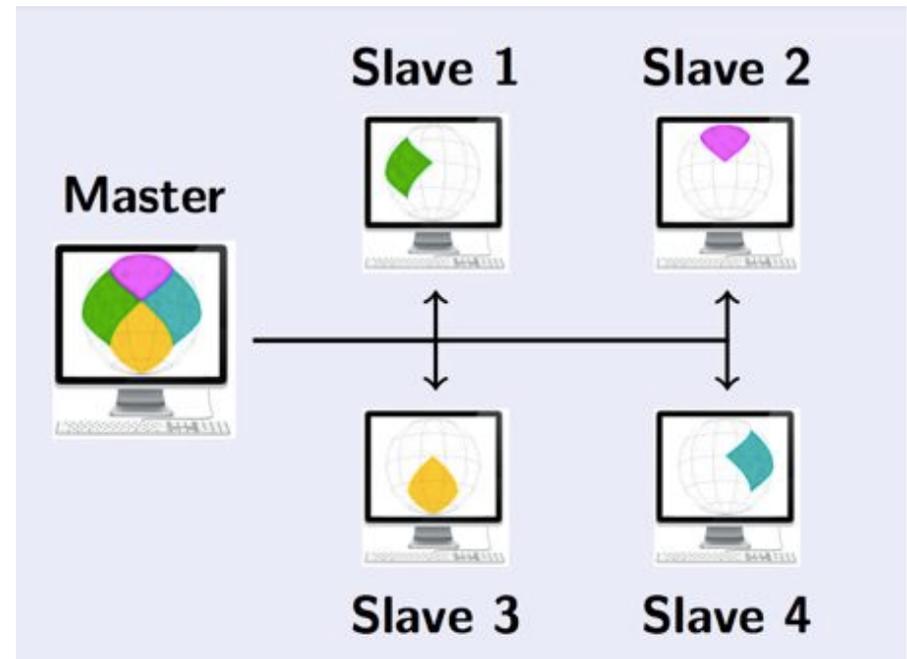
□ CDS XMatch (3)

- ... but « organized »



□ Distributed XMatch

- In the case of Hadoop / Spark, the data is distributed on clusters of servers
- Points of interest: how is the data distributed ?, how to optimize this distribution ?



□ Why this study ?...

- We are **evaluating** what Hadoop / Spark could bring
 - What it could **replace or improve**, especially in the frame of scalability (larger datasets, hardware facilities, deployments, etc.)
 - And for **which cost** (money, manpower, performances up?/down?)

- **Learn to use Hadoop / Spark**



□ Discussion

- Who is (or has tested) using Hadoop, Spark ?
 - Which use case ?
 - In production ?
 - Feedback ?
- In the frame of « bringing the code to the data » ?
 - Example: i can send a jar file to the master node to be executed « near the data »