

# 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



Astronomy ESFRI & Research Infrastructure Cluster

# □ Outlines

Context

Motivation

The data and the « cross-match » service

Test beds

Study phase

(current) Results

Conclusion and perspectives

## □ Context

A continuous exploration of new technologies,  
especially in the « Big Data » field

During a Sydney GWS session:

- Hadoop (HDFS), Spark presentation and use cases
  - Discussion
- In the frame of “bringing the computation to the data” ?

## □ Motivation (of this study)

- We wished to evaluate what Hadoop / Spark could bring by studying an appropriate use case, the « cross-match » of source catalogues:
  - Improvement of the existing service, the up to scale capability (data volumes, hardware, deployments, etc.)
  - For which cost (budget, manpower, performances (better ?))

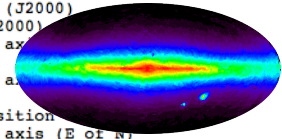
# □ Which data ?

- Data from source catalogues
- Examples (number of sources):
  - 2MASS<sup>1</sup>, 470,992,970
  - SDSS<sup>2</sup> DR9, 469,053,874

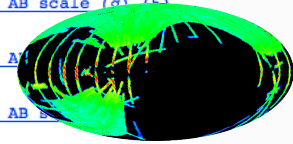
Example of a ReadMe file associated to source catalogues available through the VizieR service

<sup>1</sup>2MASS, Two Micron All Sky Survey,  
<sup>2</sup>SDSS, Sloan Digital Sky Survey

Bytes	Format	Units	Label	Explanations
1- 10	F10.6	<a href="#">deg</a>	RAdeg	(ra) Right ascension (J2000)
12- 21	F10.6	<a href="#">deg</a>	DEdeg	(dec) Declination (J2000)
23- 26	F4.2	<a href="#">arcsec</a>	errMaj	(err_maj) Semi-major axis error ellipse
28- 31	F4.2	<a href="#">arcsec</a>	errMin	(err_min) Semi-minor axis error ellipse
33- 35	I3	<a href="#">deg</a>	errPA	[0,180] (err_ang) Position angle ellipse major axis (E or W)
37- 53	A17	---	2MASS	( <a href="#">designation</a> ) <a href="#">Source designation</a> (1)



Bytes	Format	Units	Label	Explanations
1	I1	---	mode	[1,2] 1: primary (469,053,874 sources), 2: secondary (324,960,076 sources).
2	A1	---	q_mode	[+] '+' indicates clean photometry (105,969,748 sources with mode 1+)
3	I1	---	c1	<a href="#">Type (class) of object (3=galaxy, 6=star)</a> (1)
5- 23	A19	---	SDSS9	SDSS-DR9 name, based on J2000 position
24	A1	---	m_SDSS9	[*] The asterisk indicates that 2 different SDSS objects share the same SDSS9 name
27- 47	A21	---	SDSS-ID	[0-9 -] <a href="#">SDSS object identifier</a> (2)
49- 67	I19	---	objID	<a href="#">SDSS unique object identifier</a> (2)
70- 84	A15	---	Sp-ID	<a href="#">Spectroscopic Plate-MJD-Fiber identifier</a> (7)
86-104	I19	---	SpObjID	<a href="#">Pointer to the spectrum of object, or 0</a> (7)
106-124	I19	---	parentID	<a href="#">Pointer to parent (if object deblended)</a>
126-141	A16	---	flags	[0-9A-F] <a href="#">Photo Object Attribute flags</a> (3)
143-150	A8	---	Status	[0-9A-F] <a href="#">Hexadecimal status</a> (4)
153-162	F10.6	<a href="#">deg</a>	RAdeg	Right Ascension of the object (ICRS) (ra)
163-172	F10.6	<a href="#">deg</a>	DEdeg	Declination of the object (ICRS) (dec)
174-178	F5.3	<a href="#">arcsec</a>	e_RAdeg	Mean error on RAdeg (raErr)
180-184	F5.3	<a href="#">arcsec</a>	e_DEdeg	Mean error on DEdeg (decErr)
186-194	F9.4	<a href="#">yr</a>	ObsDate	Mean Observation date
196	I1	---	Q	[0/5] <a href="#">Quality of the observation (0=unknown):</a> 1=bad 2=acceptable 3=good 4=missing 5=hole (6)
198-203	F6.3	<a href="#">mag</a>	umag	? <a href="#">Model magnitude in u filter, AB scale (u)</a> (5)
204	A1	---	---	[:]
205-209	F5.3	<a href="#">mag</a>	e_umag	? Mean error on umag (err_u)
211-216	F6.3	<a href="#">mag</a>	gmag	? <a href="#">Model magnitude in g filter, AB scale (g)</a> (5)
217	A1	---	---	[:]
218-222	F5.3	<a href="#">mag</a>	e_gmag	? Mean error on gmag (err_g)
224-229	F6.3	<a href="#">mag</a>	rmag	? <a href="#">Model magnitude in r filter, AB scale (r)</a> (5)
230	A1	---	---	[:]
231-235	F5.3	<a href="#">mag</a>	e_rmag	? Mean error on rmag (err_r)
237-242	F6.3	<a href="#">mag</a>	imag	? <a href="#">Model magnitude in i filter, AB scale (i)</a> (5)
243	A1	---	---	[:]
244-248	F5.3	<a href="#">mag</a>	e_imag	? Mean error on imag (err_i)
250-255	F6.3	<a href="#">mag</a>	zmag	? <a href="#">Model magnitude in z filter, AB scale (z)</a> (5)
256	A1	---	---	[:]
257-261	F5.3	<a href="#">mag</a>	e_zmag	? Mean error on zmag (err_z)



# Which data ? (in VizieR)



VizieR Search Page

Target Name (resolved by [Sesame](#)) or Position:  J2000  2  arcmin

Target dimension:  Radius  Box size

2MASS All-Sky Catalog of Point Sources (Cutri+ 2003)

The Point Source catalogue of 470,992,970 sources. [Pages. \(470992970 rows\)](#)

VizieR Result Page

The 3 columns in **color** are computed by VizieR, and are **not part of the original data**.

[2MASS All-Sky Catalog of Point Sources \(Cutri+ 2003\)](#)

The Point Source catalogue of 470,992,970 sources. Please [acknowledge the usage of the 2MASS All-Sky Survey](#); see also the [2MASS Pages](#). Note that the magnitudes in red correspond to low quality results (upper limits or very poor photometry) (470992970 rows)

Query by **Constraints** applied on Columns (Output Order: + -)

Search Criteria

Keywords: II/246/out

Constraints

RAJ2000 deg (ra) Right ascension (J2000)

DEJ2000 deg (dec) Declination (J2000)

errMaj arcsec (err\_maj) Semi-major axis

errMin arcsec (err\_min) Semi-minor axis

errPA deg [0,180] (err\_ang) Position angle

2MASS (char) (designation) Source designation

Jmag mag <sup>(n)</sup>(j\_m) J selected default

Jcmsig mag <sup>(n)</sup>(j\_cmsig) J default magnitude

e\_Jmag mag <sup>(n)</sup>(j\_msigcom) J total magnitude

Jsnr (j\_snr) J Signal-to-noise ratio

Hmag mag <sup>(n)</sup>(h\_m) H selected default

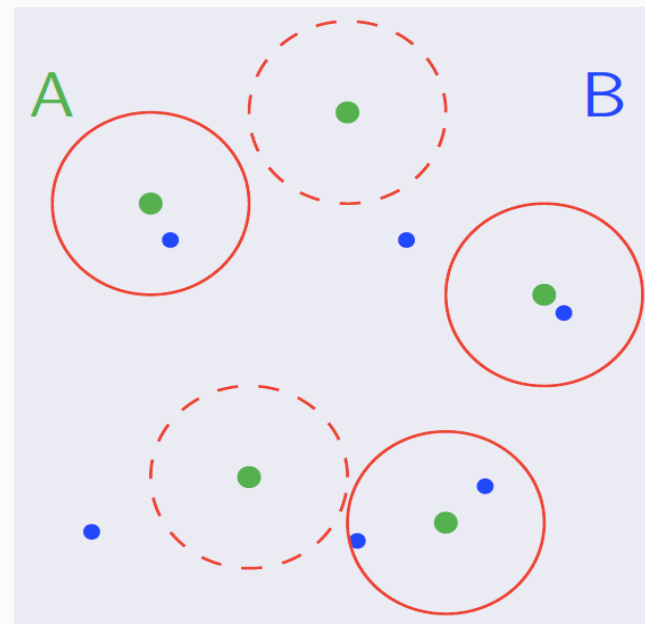
Full	r	RAJ2000	DEJ2000	RAJ2000	DEJ2000	2MASS	Jmag	e_mag	Hmag	e_mag	Kmag	e_mag	Qflg	Rflg	Bflg	Cflg	Xflg	Aflg
	arcmin	"hms"	"dms"	deg	deg		mag		mag		mag							
1	0.0171	00 42 44.337	+41 16 08.53	010.684737	+41.269035	00424433+4116085	9.453	0.052	8.668	0.051	8.475	0.051	EEE	222	111	000	2	0
2	0.0568	00 42 44.033	+41 16 06.91	010.683469	+41.268585	00424403+4116069	9.321		8.614		10.601	0.025	UUU	002	001	00c	2	0
3	0.0643	00 42 44.558	+41 16 10.38	010.685657	+41.269550	00424455+4116103	10.773	0.069	8.532		8.254		UUU	200	200	c00	2	0
4	0.0659	00 42 44.646	+41 16 09.21	010.686026	+41.269226	00424464+4116092	9.299		8.606		10.119	0.056	UUU	002	001	00c	2	0
5	0.0789	00 42 44.032	+41 16 10.83	010.683465	+41.269676	00424403+4116108	11.507	0.056	8.744		8.489		UUU	200	100	c00	2	0
6	0.0791	00 42 44.644	+41 16 10.67	010.686015	+41.269630	00424464+4116106	9.399		9.985	0.070	8.429		UUU	020	020	0c0	2	0
7	0.1008	00 42 44.465	+41 16 01.65	010.685270	+41.267124	00424446+4116016	12.070	0.035	9.301		9.057		UUU	206	200	c00	2	0
8	0.1014	00 42 43.983	+41 16 02.84	010.683263	+41.267456	00424398+4116028	12.136	0.040	9.226		8.994		UUU	200	100	c00	2	0
9	0.1111	00 42 44.203	+41 16 00.99	010.684180	+41.266941	00424420+4116009	10.065		9.374		11.504	0.052	UUU	002	002	00c	2	0
10	0.1160	00 42 43.772	+41 16 04.53	010.682383	+41.267925	00424377+4116045	12.446	0.061	11.753	0.063	9.075		AAU	220	110	cc0	2	0
11	0.1194	00 42 43.866	+41 16 12.40	010.682777	+41.270111	00424386+4116123	9.977		11.683	0.056	11.839	0.062	UUU	022	011	0cc	2	0
12	0.1221	00 42 44.601	+41 16 14.16	010.685837	+41.270599	00424460+4116141	9.880		12.051	0.068	8.934		UUU	020	020	0c0	2	0
13	0.1288	00 42 44.147	+41 16 00.06	010.683944	+41.266682	00424414+4116005	10.450		9.510		9.274		UUU	206	200	c00	2	0
14	0.1326	00 42 44.167	+41 16 15.24	010.684029	+41.270901	00424416+4116152	10.063		9.359		11.409	0.055	UUU	002	001	00c	2	0
15	0.1358	00 42 43.851	+41 16 01.40	010.682713	+41.267056	00424385+4116014	10.176		11.876	0.050	9.252		UUU	020	010	0c0	2	0
16	0.1392	00 42 44.979	+41 16 03.48	010.687414	+41.267632	00424497+4116034	12.371	0.036	9.627		9.379		UUU	200	100	c00	2	0
17	0.1522	00 42 44.843	+41 16 14.57	010.686846	+41.270714	00424484+4116145	12.872	0.061	9.433		9.178		UUU	200	200	c00	2	0
18	0.1538	00 42 44.871	+41 16 00.58	010.686963	+41.266827	00424487+4116005	10.450		12.094	0.033	11.728	0.039	UUU	622	021	bcc	2	0
19	0.1606	00 42 45.027	+41 16 13.09	010.687611	+41.270302	00424502+4116130	13.055	0.109	9.504		9.246		UUU	200	200	c00	2	0
20	0.1947	00 42 45.265	+41 16 12.54	010.688605	+41.270149	00424526+4116125	12.896	0.071	9.732		9.480		UUU	200	100	c00	2	0
21	0.2038	00 42 43.804	+41 16 18.20	010.682517	+41.271721	00424380+4116181	12.933	0.052	9.900		11.862	0.061	AAU	202	201	0cc	2	0
22	0.2085	00 42 43.450	+41 15 59.88	010.681043	+41.266632	00424345+4115598	10.511		9.815		11.861	0.049	UUU	002	001	00c	2	0
23	0.2248	00 42 43.819	+41 15 55.30	010.682581	+41.265362	00424381+4115553	10.643		9.954		12.833	0.138	UUU	002	002	00c	2	0
24	0.2377	00 42 43.124	+41 16 03.21	010.679682	+41.267558	00424312+4116032	10.684		9.985		12.268	0.051	UUU	002	001	00c	2	0

Example: 2MASS and a research around Andromeda

# □ ...and the CDS « cross-match » service

- The « cross-match » service does a cross correlation of sources between (very) large catalogues (current size:  $10^9$ ).

Fuzzy join between 2 tables of several hundred millions of lines



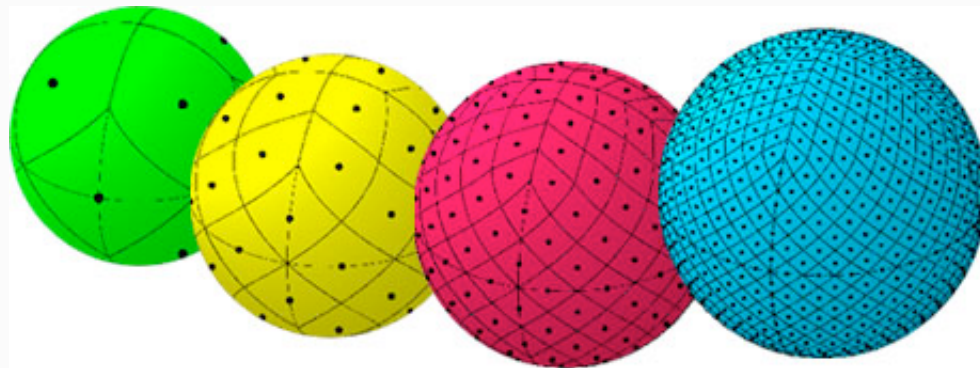
## □ ...and the CDS « cross-match » service (2)

- It is possible to do it with catalogues proposed by the CDS but also to upload your own data (a table with positions) to cross-match it with these catalogues.
- It is based on optimized developments and implemented on a well-sized server (enabling on-line use).



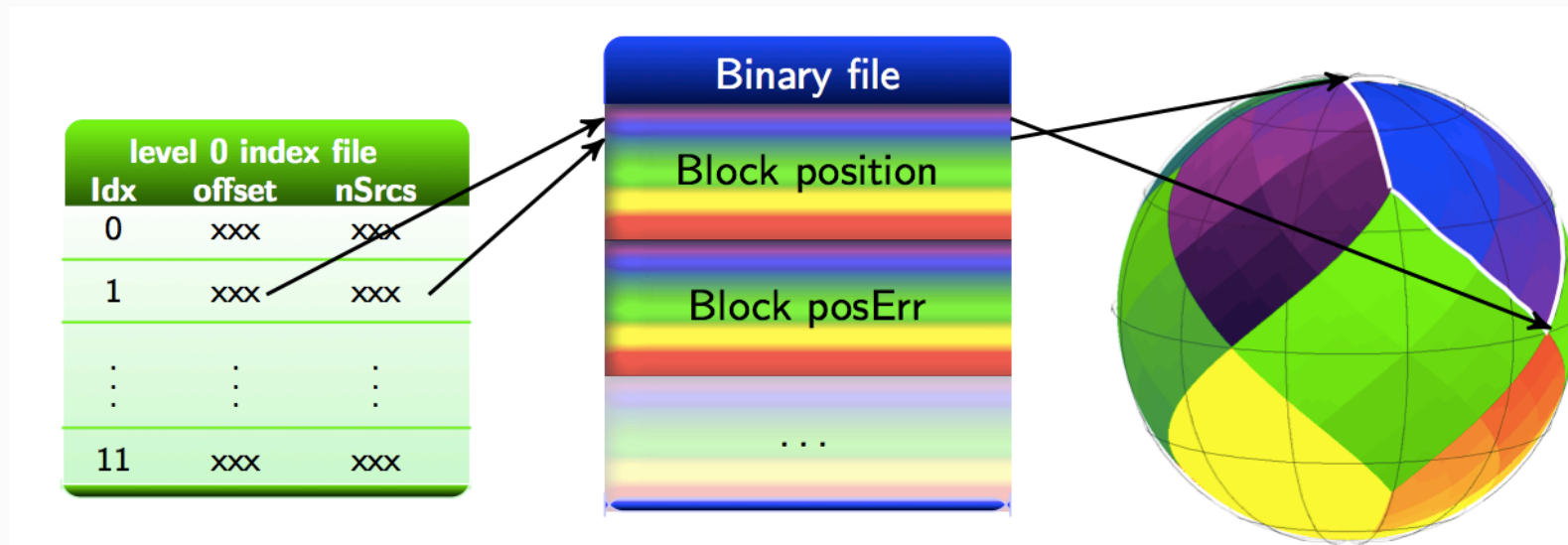
## □ ...and the CDS « cross-match » service (3)

- Which area ?
  - Full sky: all the sources
  - A cone: only the sources which are at a certain angular distance from a given position
  - A HEALPix cell



# □ ...and the CDS « cross-match » service (4)

- Data is not distributed but organised and stored on one server



The sky is cut into diamonds of the same size, pixels, each source or sky object is a numbered pixel.

# Illustrations

## Example: X-Match of 2MASS & SDSS DR9

The screenshot shows the CDS X-Match Service interface. At the top, there's a navigation bar with 'Portal', 'Simbad', 'VizieR', 'Aladin', 'X-Match', 'Other', and 'Help'. Below that, the 'CDS X-Match Service' header includes 'X-match', 'Tables management', and 'Documentation' buttons, along with 'Login', 'Preferences', and 'Register' links.

The main section is titled 'Choose tables to cross-match'. It features two input fields: '2MASS' and 'SDSS DR9'. Below these are buttons for 'VizieR', 'SIMBAD', and 'My store'. Two table cards are displayed: '2MASS All-Sky Catalog of Point Sources (Cutri+ 2003)' with 470,992,970 rows and 'The SDSS Photometric Catalog, Release 9 (Adelman-McCarthy+, 2012)' with 794,013,950 rows.

Under 'Cross-match criteria', 'By position' is selected. The 'Radius' is set to 5 arcsec. 'By position including error' is also an option, with 'Sigma' set to 3.43935 (completeness: 99.73%) and 'Max. distance' set to 5 arcsec. Under 'Cross-match area', 'All sky' is selected. The 'Center' is set to 'Position/Object name' and 'Radius' is set to 'deg'. 'Healpix cell (ICRS, NESTED scheme)' is also an option, with 'Nside' set to 4 and 'Index' set to 0.

A 'Begin the X-Match' button is visible at the bottom left of the main section. A date '11/05/2016' is shown at the bottom left of the entire screenshot.

Two inset windows show the job management interface. The top inset shows a job in progress: '2MASS' and 'SDSS DR9' with 'fixed radius' options, begun on '06/04/2016 at 10:21', with a status of 'executing' and an 'Abort' button. The bottom inset shows the same job as 'completed' on '06/04/2016 at 10:21', with a 'Get result' button and a dropdown menu for 'Download as CSV', 'Download as ASCII', and 'Download as VOTable'.

11/05/2016

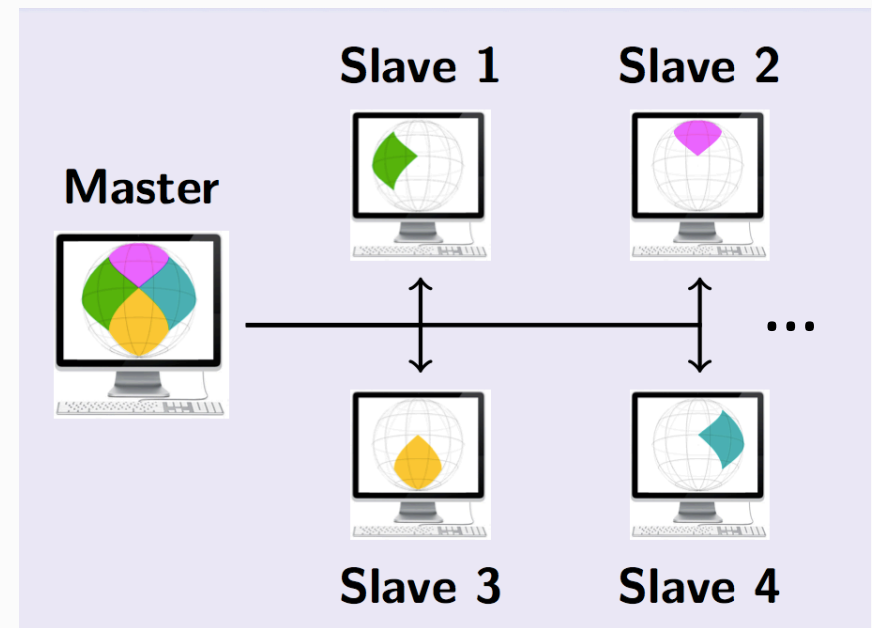
# Illustrations (2)

**Example:  
An “excerpt”  
of the result  
in CSV**

```
Téléchargements — vi 1459930879752A.csv — 108x36
angDist,2MASS,RAJ2000,DEJ2000,errHalfMaj,errHalfMin,errPosAng,Jmag,Hmag,Kmag,e_Jmag,e_Hmag,e_Kmag,Qfl,Rfl,X,
MeasureJD,SDSS9,RAdeg,DEdeg,errHalfMaj,errHalfMin,errPosAng,umag,gmag,rmag,imag,zmag,e_umag,e_gmag,e_rmag,e_
imag,e_zmag,objID,cI,q_mode,flags,Q,obsDate,pmRA,e_pmRA,pmDE,e_pmDE,SpObjID,zsp,e_zsp,f_zsp,spType,spCl,subC
lass
0.305453,02595905+0000200,44.996055,+0.005565,0.170,0.160,76,16.376,15.770,15.258,0.097,0.140,0.141,ABB,222,
0,2451084.8062,J025959.06+000020.2,44.996116,+0.005624,0.002,0.002,90,19.548,18.186,17.619,17.379,17.241,0.0
28,0.006,0.007,0.007,0.013,1237663784217084122,6,1,0000201090020010,3,2003.8857,13,3,-5,3,0,,,,,
0.080507,03000116+0001113,45.004857,+0.019806,0.060,0.060,90,12.529,11.954,11.874,0.024,0.030,0.029,AAA,222,
0,2451084.8062,J030001.17+000111.2,45.004879,+0.019802,0.061,0.060,90,17.398,15.191,14.183,16.934,13.777,0.0
11,0.005,0.003,0.018,0.006,1237663784217083948,6,0,0000F81090060010,3,2003.8857,20,4,24,4,0,,,,,
1.331290,03000116+0001113,45.004857,+0.019806,0.060,0.060,90,12.529,11.954,11.874,0.024,0.030,0.029,AAA,222,
0,2451084.8062,J030001.08+000110.8,45.004509,+0.019681,0.062,0.057,0,24.566,25.148,17.596,13.890,22.827,2.39
3,1.716,0.032,0.001,2.226,1237663784217083950,3,0,0001F80092061110,3,2003.8857,,,,,0,,,,,
4.789590,03000116+0001113,45.004857,+0.019806,0.060,0.060,90,12.529,11.954,11.874,0.024,0.030,0.029,AAA,222,
0,2451084.8062,J030001.01+000115.5,45.004220,+0.020974,0.002,0.002,90,21.956,19.689,18.110,16.886,16.261,0.1
41,0.014,0.008,0.006,0.008,1237663784217083949,6,1,0000201812060010,3,2003.8857,,,,,0,,,,,
0.116926,03000100+0001154,45.004193,+0.020956,0.060,0.060,90,14.845,14.223,14.016,0.056,0.077,0.055,AAA,222,
0,2451084.8062,J030001.01+000115.5,45.004220,+0.020974,0.002,0.002,90,21.956,19.689,18.110,16.886,16.261,0.1
41,0.014,0.008,0.006,0.008,1237663784217083949,6,1,0000201812060010,3,2003.8857,,,,,0,,,,,
4.728929,03000100+0001154,45.004193,+0.020956,0.060,0.060,90,14.845,14.223,14.016,0.056,0.077,0.055,AAA,222,
0,2451084.8062,J030001.08+000110.8,45.004509,+0.019681,0.062,0.057,0,24.566,25.148,17.596,13.890,22.827,2.39
3,1.716,0.032,0.001,2.226,1237663784217083950,3,0,0001F80092061110,3,2003.8857,,,,,0,,,,,
4.833083,03000100+0001154,45.004193,+0.020956,0.060,0.060,90,14.845,14.223,14.016,0.056,0.077,0.055,AAA,222,
0,2451084.8062,J030001.17+000111.2,45.004879,+0.019802,0.061,0.060,90,17.398,15.191,14.183,16.934,13.777,0.0
11,0.005,0.003,0.018,0.006,1237663784217083948,6,0,0000F81090060010,3,2003.8857,20,4,24,4,0,,,,,
0.084417,02595132+0002369,44.963851,+0.043587,0.220,0.170,95,16.476,16.057,15.564,0.113,0.175,BCU,220,0,245
1084.8062,J025951.33+000236.9,44.963874,+0.043591,0.002,0.002,90,20.998,18.942,18.088,17.765,17.573,0.071,0.
009,0.007,0.008,0.016,1237663784217084100,6,1,0000001010000000,3,2003.8857,6,3,-3,3,0,,,,,
0.267343,02595881+0002175,44.995074,+0.038204,0.380,0.310,0,16.746,15.814,16.125,0.134,0.140,0.324,BBD,222,0
,2451084.8062,J025958.80+000217.3,44.995029,+0.038145,0.009,0.008,90,24.543,21.773,20.167,18.857,18.180,0.69
0,0.055,0.021,0.012,0.024,1237663784217084474,6,1,0000001010000000,3,2003.8857,0,5,1,5,0,,,,,
0.120901,03001158+0002539,45.048281,+0.048329,0.180,0.070,0,13.354,12.874,12.699,0.025,0.030,0.030,AAA,222,0
,2451813.9014,J030011.58+000253.8,45.048268,+0.048298,0.001,0.000,90,17.725,15.789,14.967,14.983,14.450,0.01
1,0.004,0.005,0.001,0.005,1237663784217084036,6,0,0001981294061048,3,2003.8857,-4,3,-15,3,0,,,,,
@
```

# □ Distribution ?

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



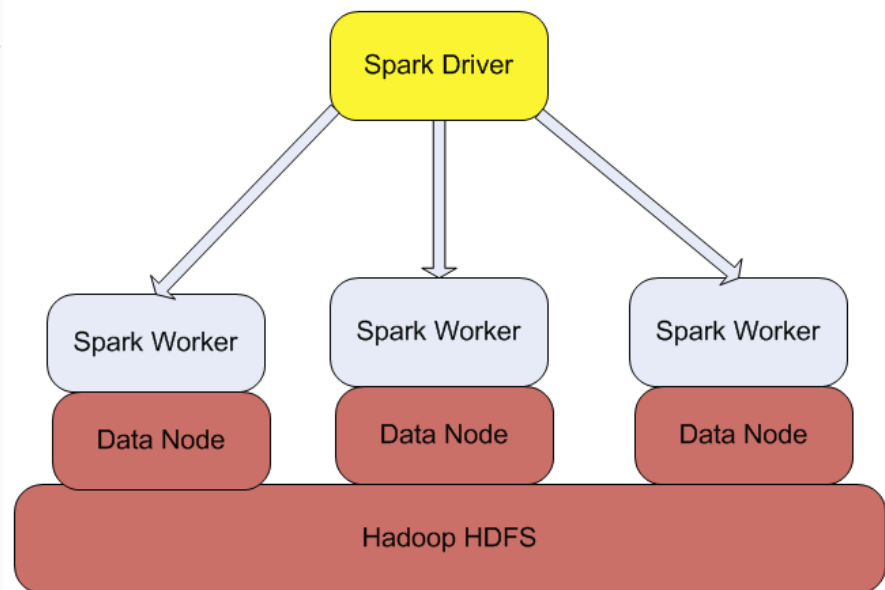
# □ Test beds

- Data: >10k catalogues (SDSS, 2MASS, etc.)
  - Up to ~ 60 GB and several millions of elements on the output side (examples: 2MASS 58GB, SDSS DR9 54GB, ~49  $10^6$  output elements)
- Internal resources
  - Up to 6 nodes (4 cores, 16GB, 1 TB), common desktop machines under Ubuntu 14.04
- External resources from a provider
  - 12 nodes (dedicated servers, 4 cores, 32GB, Raid 2\*2TB), Ubuntu 14.04

X-Match server (2\*6 cores, 32GB, 12TB (15k tours))

## □ Test beds (2)

- Common architecture using the Apache distributions (Spark 1.5.0 for Hadoop 2.6) + Java (no Cloudera, Hortonworks, ...)
- Standalone mode (with Spark own's cluster manager)
  - Without Apache Yarn, Mesos, ...
  - Quick add of new nodes



Credits : BigHadoop

# □ Study phase – Data preparation

- Before the execution the **input files are stored into HDFS.**
- These files are, in a first step, **loaded into 2 RDDs** ((Resilient Distributed Dataset, a distributed data collection) where each line of the RDD is an element containing the information about an object in the Sky.
- **Each RDD is then transformed in a PairRDD** (RDD containing a (Key, Value) pair): a **key** representing the **source pixel number** is attributed to each element of the RDD based on the HEALPix tessellation of the Sky.
- The elements of the PairRDDs are then **(Key, Value)** couples where the Value contains all the information whose the source (ra, dec) coordinates in the equatorial system.



## □ Study phase – Data preparation (2)

- The **PairRDDs** are then **distributed** over the **cluster nodes**.
- This **distribution** is done following a **hash partitioning** where the PairRDDs are split in partitions which will be stored over the nodes.
- The hash partitioning consists in **grouping all the elements having the same key (same pixel number) in the same partition**.
- The partitions are stored into different nodes
  - **The elements with the same key are on the same nodes**
  - This data distribution is essential to the second programme phase.
- At the end the **PairRDDs** are stored **into HDFS as binary files following a method preserving the structure (Key, Value)**.

## □ Study phase – Join

- The binary files stored previously are directly loaded into two PairRDDs.
- A treatment is applied on the second PairRDD: a **duplication** of some **sources** in the **neighbour pixels**.
- The **2 PairRDDs** are then **joined following the Key**. This join generates a new PairRDD where the elements are (Key, Value1, Value2) triples.
- As the join is done following the Key (cell number), 2 near sources can be in the different cells and so they are not joined (=> **duplication of sources in the neighbour cells to limit the side effects**).

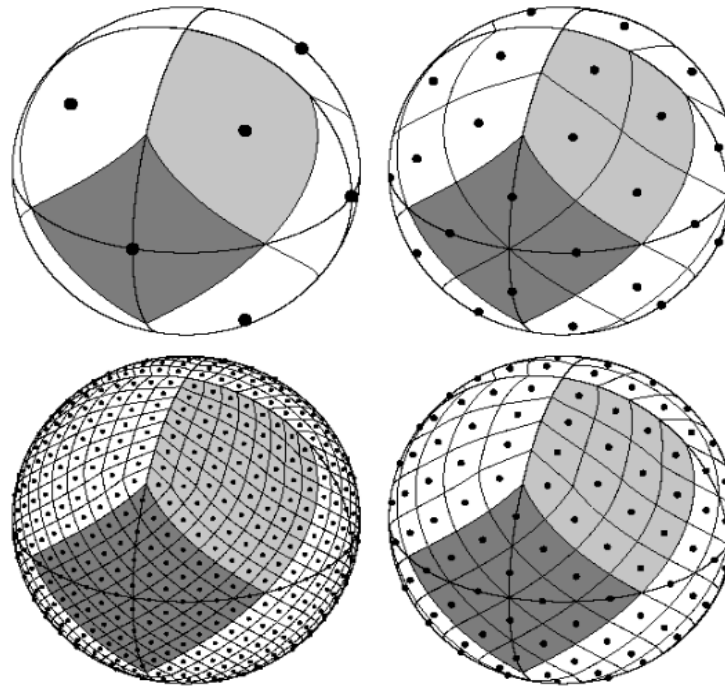
## □ Study phase – Join (2)

- The **duplication** steps
  - A circle with a fixed radius is drawn around the source
  - If neighbour pixels are partially in this circle, the source is then duplicated in the neighbour cells.
- The **joined elements** are then **filtered**
  - Only the joined elements which distance between the two sources is under a **given threshold** are conserved.
- The **final result is stored in HDFS** in a text format for a later visualization and use.

# □ Illustration

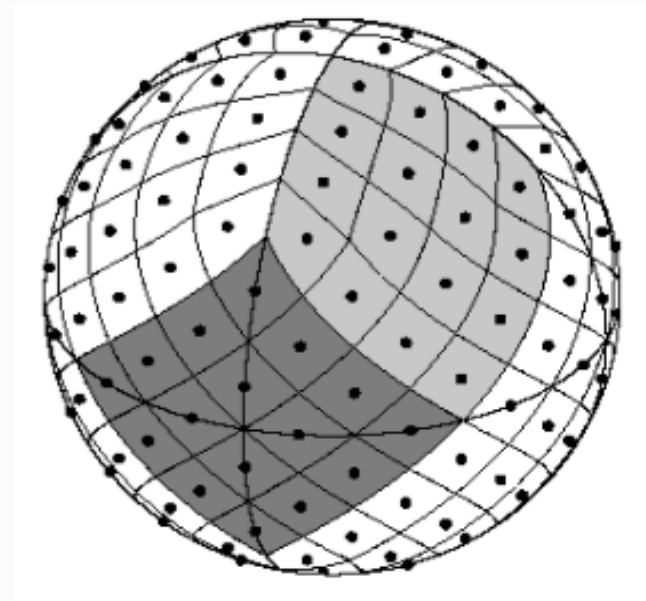
- A X-Match implementation in MapReduce, Couples (Key = pixel number, Value)

HEALPix sky cutting



## □ Illustration (2)

- Side effects
  - Fuzzy join
  - Source duplication in the neighbour cells if needed



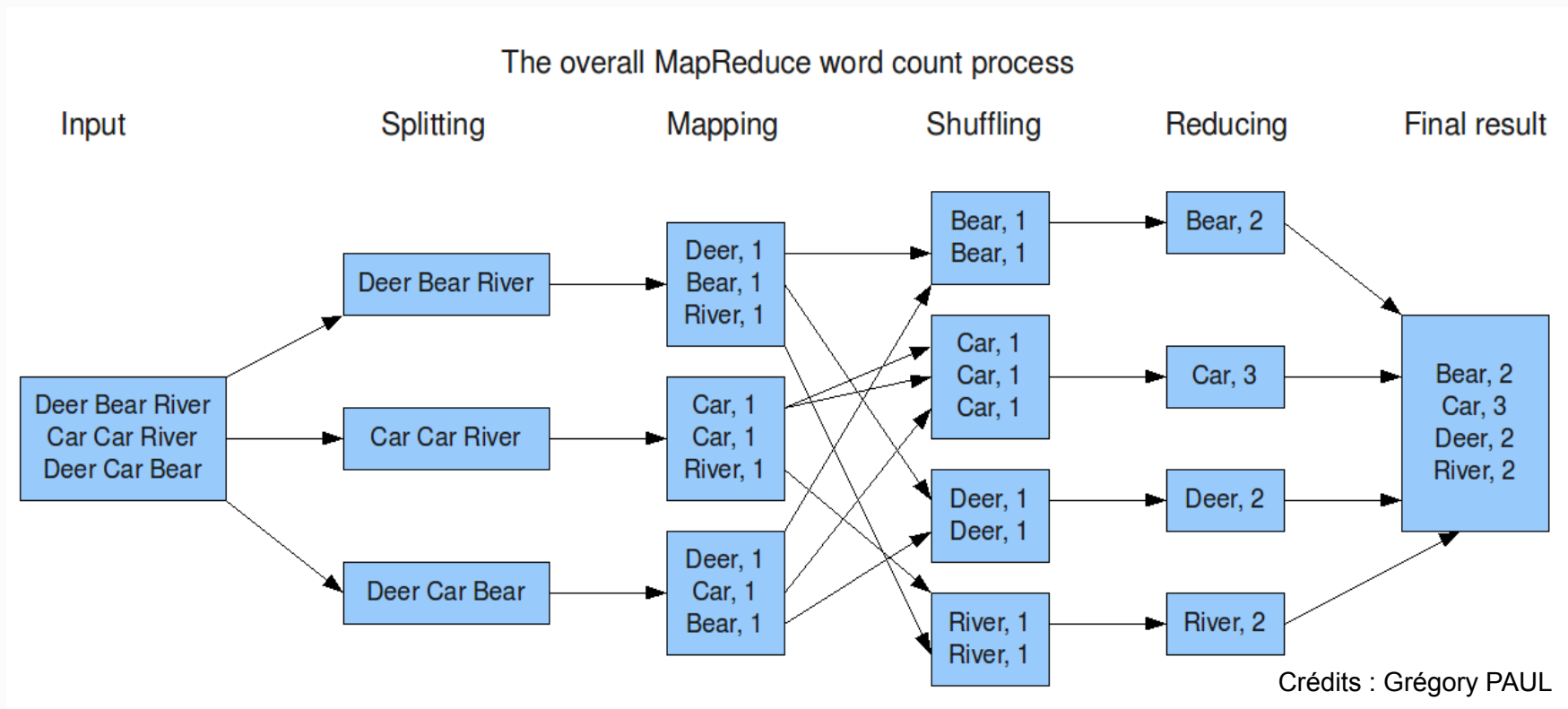
Credits : HEALPix – [arXiv:astro-ph/0409513](https://arxiv.org/abs/astro-ph/0409513)

## □ Study phase – Co-location

- During the hash partitioning of the RDDs, the elements having the same key are stored on the same nodes for a given RDD.
- **This means not that the keys which are common to 2 RDDs are also on the same nodes.** In this case, it implies a transfer **overhead** between the nodes during the join, which has an impact on the **performances** (=> Shuffle + no co-location of the Data).

# ☐ « Shuffle » phase

- Re-distribution on the nodes



# □ Results

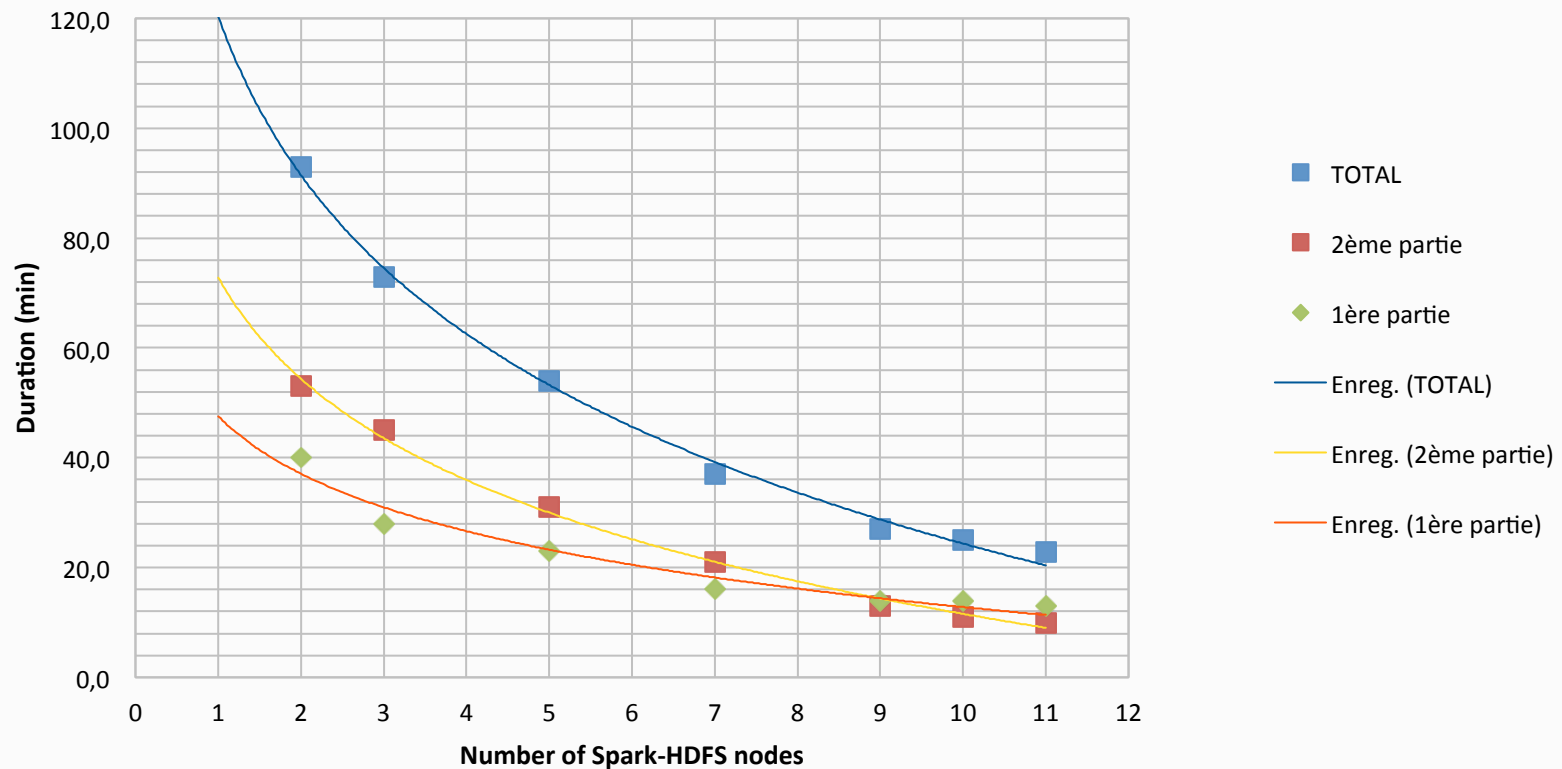
- Input data (SDSS DR7 (primary sources) and 2MASS): 54GB and 58GB file size; 357 175 411 and 470 992 970 elements
- Output data: 49 208 820 elements

Cross-Match (duplication des sources faite dans la 2e partie ; avec toutes les données en sortie)											
Taille des blocs HDFS = 128MB pour les fichiers en entrée ; sdss7.csv et 2mass.csv répliqués 2x											
HashPartitioner	60 partitions										
Taille des blocs HDFS en sortie	32MB										
Nombre de nœuds Spark/HDFS	1	2	3	4	5	6	7	8	9	10	11
<b>1ère partie : préparation des données</b>		40,0	28,0		23,0		16,0		14,0	14,0	13,0
mapToPair (sdss7.csv)		7,8			5,1		4,9		4,9	4,8	4,7
saveAsHadoopFile (sdss7.bin)		10,0			5,7		2,7		2,0	2,3	1,5
mapToPair (2mass.csv)		8,5			5,7		5,2		5,2	5,1	5,0
saveAsHadoopFile (2mass.bin)		13,0			6,5		3,6		1,9	1,6	1,4
<b>2ème partie : jointure</b>		53,0	45,0		31,0		21,0		13,0	11,0	9,9
mapToPair (sdss7.bin)					7,2		4,7		3,5	3,0	2,9
flatMapToPair (2mass.bin)					11,8		8,3		5,5	4,9	4,3
saveAsTextFile (crossMatch_D.txt)					12,0		7,6		3,4	2,4	2,3
<b>TOTAL</b>		93,0	73,0		54,0		37,0		27,0	25,0	22,9



# Results (2)

X-Match duration / number of nodes



The CDS X-Match service needs 15 minutes of computation for the same data (which is split in multiple files in HDFS), It corresponds to the second part (data is already prepared)

# □ Conclusion and perspective

- Our current results:
  - We have reached an execution time better than the X-Match service
  - From 8 nodes it could be an alternative to the current architecture
  - Concerning the cost the dedicated server set (rent) is interesting (example:  $8 \times 60 \times 12$ , around 6000 euros / an)
- Remark: following recent discussions, the test beds could be optimized (RAM per core, no RAID, etc.)

## □ Conclusion and perspective (2)

- Bottleneck: « shuffle »
  - Optimisation (?) of the code using the « data co-location », « block affinity groups » is an on-going work at Apache
- We wish to do new tests with more nodes but also with other configurations (RAM, Hard disks, etc.)
- On going collaboration with Julien Nauroy (Université Paris Sud) who has deployed a Spark architecture in his University

# □ Links

- Apache Spark, <http://spark.apache.org/>
- Apache Hadoop, <http://hadoop.apache.org/>
- Spark : Cluster Computing with Working Sets, Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica, University of California, Berkeley,  
[http://static.usenix.org/legacy/events/hotcloud10/tech/full\\_papers/Zaharia.pdf](http://static.usenix.org/legacy/events/hotcloud10/tech/full_papers/Zaharia.pdf)
- Optimizing Shuffle Performance in Spark, Aaron Davidson, Andrew Or, UC Berkeley,  
[http://www.cs.berkeley.edu/~kubitron/courses/cs262a-F13/projects/reports/project16\\_report.pdf](http://www.cs.berkeley.edu/~kubitron/courses/cs262a-F13/projects/reports/project16_report.pdf)
- Resilient Distributed Datasets : A Fault-Tolerant Abstraction for In-Memory Cluster Computing, Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica, University of California, Berkeley,  
[https://www.cs.berkeley.edu/~matei/papers/2012/nsdi\\_spark.pdf](https://www.cs.berkeley.edu/~matei/papers/2012/nsdi_spark.pdf)
- JavaSpark Api, <http://spark.apache.org/docs/latest/api/java/>
- HEALPix, <http://healpix.jpl.nasa.gov/>