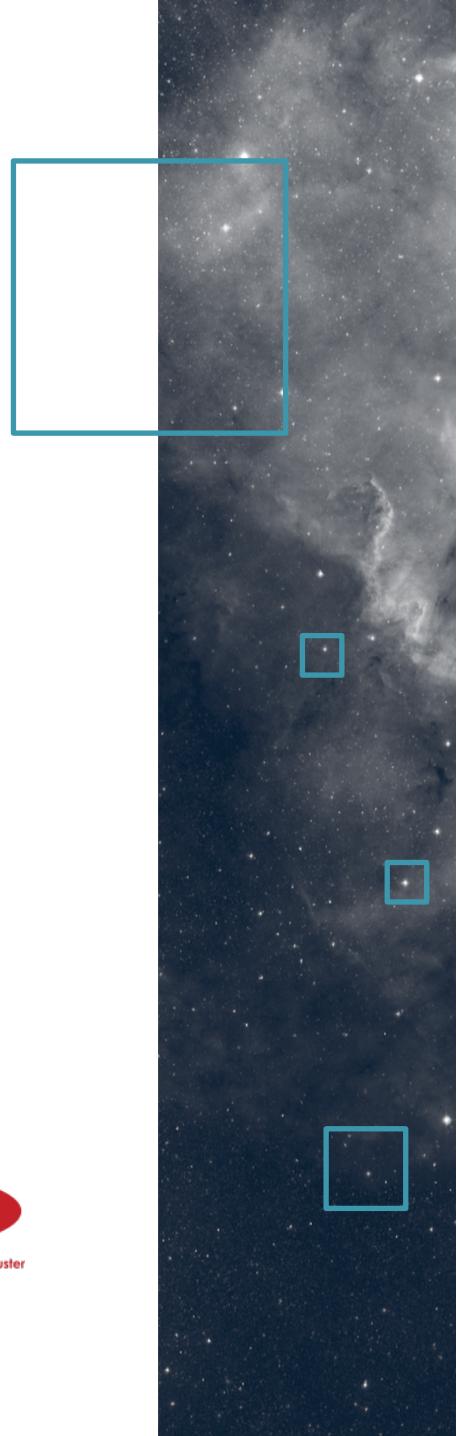


# Application en astrophysique, « cross-match » de catalogues de sources



André Schaaff, François-Xavier Pineau

CDS, Centre de Données astronomiques de Strasbourg

Noémie Wali

Elève-ingénierie UTBM, Université de technologie de Belfort-Montbéliard

9<sup>ème</sup> Journée Loops

**Apache Spark : la distribution de calculs selon Hadoop, LAL Orsay, 7 avril 2016**



CENTRE DE DONNÉES  
ASTRONOMIQUES DE STRASBOURG





# Le fil d'Ariane

Contexte

La motivation

Les données et le service de « cross-match »

Les bancs de test

Phase d'étude

Résultats (actuels)

Conclusion et perspectives



## □ Contexte

Une exploration continue de nouvelles technologies, notamment « Big Data »  
Pas de compétence particulière au niveau  
Hadoop / Spark



# La motivation (de cette étude)

- Nous souhaitions évaluer ce que Hadoop / Spark pouvait apporter en étudiant un cas d'utilisation précis, le « cross-match » de catalogues de sources:
  - Remplacement ou amélioration de l'existant, notamment au niveau du passage à l'échelle (jeux de données de taille croissante (exponentielle...), souplesse au niveau matériel, déploiements, etc.)
  - Pour quel coût (budget, main d'oeuvre, performances (améliorées ?))
- Mais également nous familiariser avec Hadoop / Spark



# Les données...

- Données issues des catalogues de sources
- Exemples (nb sources):
  - 2MASS<sup>1</sup>, 470,992,970
  - SDSS<sup>2</sup> DR9, 794,013,950

Exemple de fichiers ReadMe associés aux catalogues de sources accessibles via le service VizieR

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

<sup>2</sup>SDSS, Sloan Digital Sky Survey

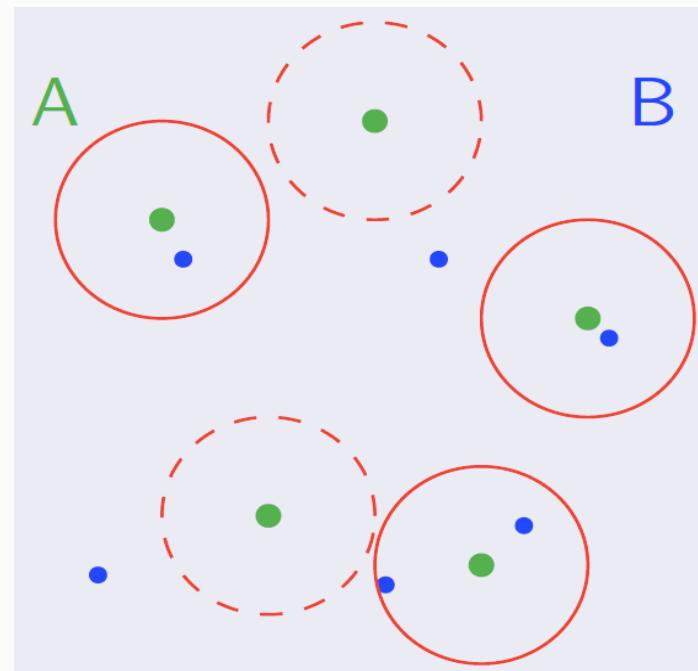
Bytes	Format	Units	Label	Explanations	
1- 10	F10.6	deg	RAdeg	(ra) Right ascension (J2000)	
12- 21	F10.6	deg	DEdeg	(dec) Declination (J2000) (dec)	
23- 26	F4.2	arcsec	errMaj	(err_maj) Semi-major axis of position error ellipse	
28- 31	F4.2	arcsec	errMin	(err_min) Semi-minor axis of position error ellipse	
33- 35	I3	deg	errPA	[0,180] (err_ang) Position angle of error ellipse major axis (E of N)	
37- 53	A17	---	2MASS	(designation) Source designation (1)	
55- 60	F6.3	mag	Jmag	?(j,m) J selected default magnitude (2)	
62- 66	F5.3	mag	Jcmsg	?(j,cmsg) J default magnitude uncertainty (3)	
68- 72	F5.3	mag	e_Jmag	?(j,msigcom) J total magnitude uncertainty (4)	
74- 83	F10.1	---	Jsnr	?(j,snr) J Signal-to-noise ratio	
85- 90	A6.2	---	Dmag	?(d,m) D selected default magnitude (2)	
					inty (3)
					inty (4)
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	Type (class) of object (3=galaxy, 6=star) (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 -] SDSS object identifier (2)	
49- 67	I19	---	objID	SDSS unique object identifier (2)	
70- 84	A15	---	Sp-ID	Spectroscopic Plate-MJD-Fiber identifier (7)	
86-104	I19	---	SpObjID	Pointer to the spectrum of object, or 0 (7)	
106-124	I19	---	parentID	Pointer to parent (if object deblended)	
126-141	A16	---	flags	[0-9A-F] Photo Object Attribute flags (3)	
143-150	A8	---	Status	[0-9A-F] Hexadecimal status (4)	
153-162	F10.6	deg	RAdeg	Right Ascension of the object (ICRS) (ra)	
163-172	F10.6	deg	DEdeg	Declination of the object (ICRS) (dec)	
174-178	F5.3	arcsec	e_RAdeg	Mean error on RAdeg (raErr)	
180-184	F5.3	arcsec	e_DEdeg	Mean error on DEdeg (decErr)	
186-194	F9.4	yr	ObsDate	Mean Observation date	
196	I1	---	Q	[0/5] Quality of the observation (0=unknown): 1=bad 2=acceptable 3=good 4=missing 5=hole (6)	
198-203	F6.3	mag	umag	? Model magnitude in u filter, AB scale (u) (5)	
204	A1	---	---	[:]	
205-209	F5.3	mag	e_umag	? Mean error on umag (err_u)	
211-216	F6.3	mag	gmag	? Model magnitude in g filter, AB scale (g) (5)	
217	A1	---	---	[:]	
218-222	F5.3	mag	e_gmag	? Mean error on gmag (err_g)	
224-229	F6.3	mag	rmag	? Model magnitude in r filter, AB scale (r) (5)	
230	A1	---	---	[:]	
231-235	F5.3	mag	e_rmag	? Mean error on rmag (err_r)	
237-242	F6.3	mag	imag	? Model magnitude in i filter, AB scale (i) (5)	
243	A1	---	---	[:]	
244-248	F5.3	mag	e_imag	? Mean error on imag (err_i)	
250-255	F6.3	mag	zmag	? Model magnitude in z filter, AB scale (z) (5)	
256	A1	---	---	[:]	
257-261	F5.3	mag	e_zmag	? Mean error on zmag (err_z)	



## □ ...et le service de « cross-match »

- Le service de « cross-match » permet de réaliser une corrélation / identification croisée de sources entre (très) grands catalogues (ordre de grandeur actuelle:  $10^9$ ).

Jointure floue entre 2 tables de plusieurs centaines de millions de données



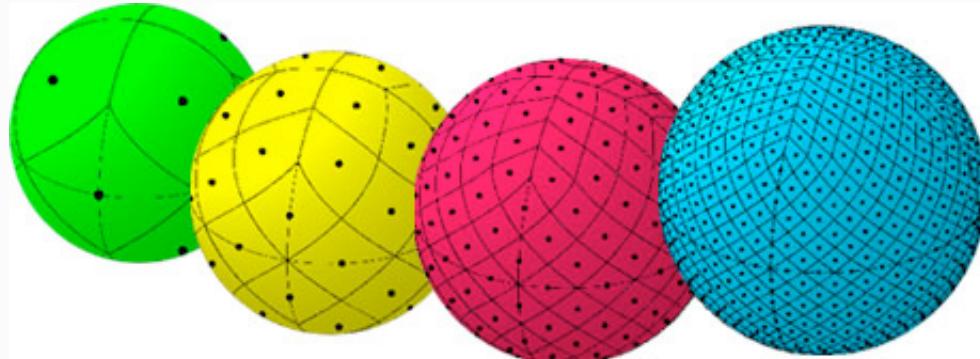


## ...et le service de « cross-match » (2)

- Il est possible de le faire pour les catalogues proposés par le CDS mais également de télécharger ses propres données (une table avec positions) pour les croiser avec l'un de ces catalogues.
- C'est un service basé sur des développements optimisés et une implémentation sur un serveur bien dimensionné (pour une utilisation en ligne).
- Par position.

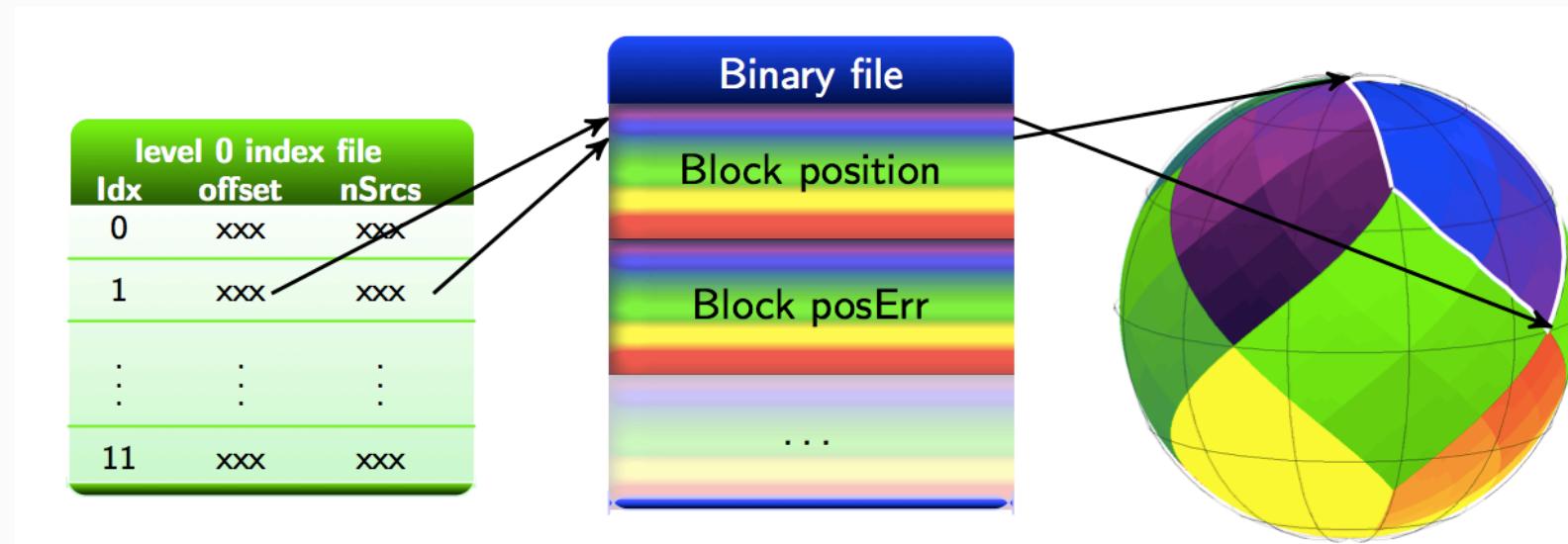
## □ ...et le service de « cross-match » (3)

- Zone concernée
  - Tout le ciel: toutes les sources
  - Un cône: uniquement les sources à une certaine distance angulaire d'une position donnée
  - Une cellule HEALPix (pixellisation du ciel)



## □ ...et le service de « cross-match » du CDS (4)

- Les données ne sont pas distribuées mais organisées et stockées sur un système RAID



Le ciel est découpé en losanges de tailles identiques, appelés pixels, chaque source ou objet du ciel est positionné dans un pixel numéroté.



# Illustrations

CDS X-Match Service X-match Tables management Documentation Login Preferences Register

Choose tables to cross-match

2MASS SDSS DR9

VizieR SIMBAD My store VizieR SIMBAD My store

2MASS All-Sky Catalog of Point Sources (Cutri+ 2003)  
470,992,970 rows

The SDSS Photometric Catalog, Release 9 (Adelman-McCarthy+, 2012)  
794,013,950 rows

Begin the X-Match

Visualize and manage your cross-match jobs

List of X-match jobs

Table 1	Table 2	Options	Begin	Status	Actions
2MASS	SDSS DR9	fixed radius +	06/04/2016 at 10:21	executing	<input type="checkbox"/> Abort

For the selected job(s):

Visualize and manage your cross-match jobs

List of X-match jobs

Table 1	Table 2	Options	Begin	Status	Actions
2MASS	SDSS DR9	fixed radius +	06/04/2016 at 10:21	completed	<input type="checkbox"/> Get result

For the selected job(s):

Begin the X-Match

7/04/2016

Detailed description: This screenshot shows the CDS X-Match Service interface. At the top, there's a navigation bar with links to Portal, Simbad, VizieR, Aladin, X-Match, Other, and Help. Below that is a sub-navigation bar for 'CDS X-Match Service' with tabs for X-match, Tables management, and Documentation, along with login and preferences links. The main area is titled 'Choose tables to cross-match' and shows two datasets: '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. Below these are sections for 'Cross-match criteria' (set to 'By position' with a radius of 5 arcsec) and 'Cross-match area' (set to 'All sky'). A large red box highlights the 'Begin the X-Match' button. To the right, there are two tables titled 'Visualize and manage your cross-match jobs'. The first table shows a single job in progress ('executing') between 2MASS and SDSS DR9 with a fixed radius. The second table shows the same job completed ('completed') with options to 'Get result', 'Download as CSV', 'Download as ASCII', or 'Download as VOTable'. A date '7/04/2016' is visible at the bottom left.

Exemple:  
X-Match de  
2MASS et SDSS DR9



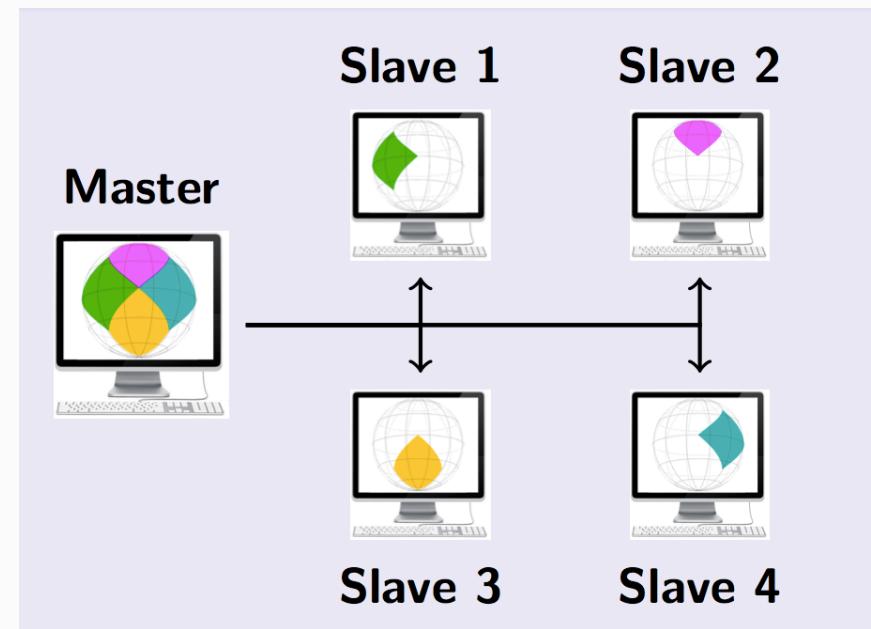
# Illustrations (2)

Exemple:  
Un « extrait » du  
résultat en CSV

```
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,cl,q_mode,flags,Q,ObsDate,pmRA,e_pmRA,pmDE,e_pmDE,SpObjID,zsp,e_zsp,f_zsp,spType,spCl,subClass
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,+0.005624,0.002,0.002,90,19.548,18.186,17.619,17.379,17.241,0.028,0.006,0.007,0.007,0.013,1237663784217084122,6,1,0000201090020010,3,2003.8857,13,3,-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,+0.019802,0.061,0.060,90,17.398,15.191,14.183,16.934,13.777,0.011,0.005,0.003,0.018,0.006,1237663784217083948,6,0,0000F81090060010,3,2003.8857,20,4,24,4,0,,,
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,+0.019681,0.062,0.057,0,24.566,25.148,17.596,13.890,22.827,2.393,1.716,0.032,0.001,2.226,1237663784217083950,3,0,0001F80092061110,3,2003.8857,,,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,+0.020974,0.002,0.002,90,21.956,19.689,18.110,16.886,16.261,0.141,0.014,0.008,0.006,0.008,1237663784217083949,6,1,0000201812060010,3,2003.8857,,,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,+0.020974,0.002,0.002,90,21.956,19.689,18.110,16.886,16.261,0.141,0.014,0.008,0.006,0.008,1237663784217083949,6,1,0000201812060010,3,2003.8857,,,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,+0.019681,0.062,0.057,0,24.566,25.148,17.596,13.890,22.827,2.393,1.716,0.032,0.001,2.226,1237663784217083950,3,0,0001F80092061110,3,2003.8857,,,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,+0.019802,0.061,0.060,90,17.398,15.191,14.183,16.934,13.777,0.011,0.005,0.003,0.018,0.006,1237663784217083948,6,0,0000F81090060010,3,2003.8857,20,4,24,4,0,,,
0.084417,02595132+0002369,44.963851,+0.043587,0.220,0.170,95,16.476,16.057,15.564,0.113,0.175,,BCU,220,0,2451084.8062,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,+0.038145,0.009,0.008,90,24.543,21.773,20.167,18.857,18.180,0.690,0.055,0.021,0.012,0.024,1237663784217084474,6,1,0000001010000000,3,2003.8857,0,5,1,5,0,,,
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,+0.048298,0.001,0.000,90,17.725,15.789,14.967,14.983,14.450,0.011,0.004,0.005,0.001,0.005,1237663784217084036,6,0,0001981294061048,3,2003.8857,-4,3,-15,3,0,,,
@
```

## □ Et s'il était distribué?

- Dans le cas de Hadoop / Spark, les données sont distribuées sur plusieurs serveurs
- Comment les données sont-elles distribuées ?, comment optimiser cette distribution ?





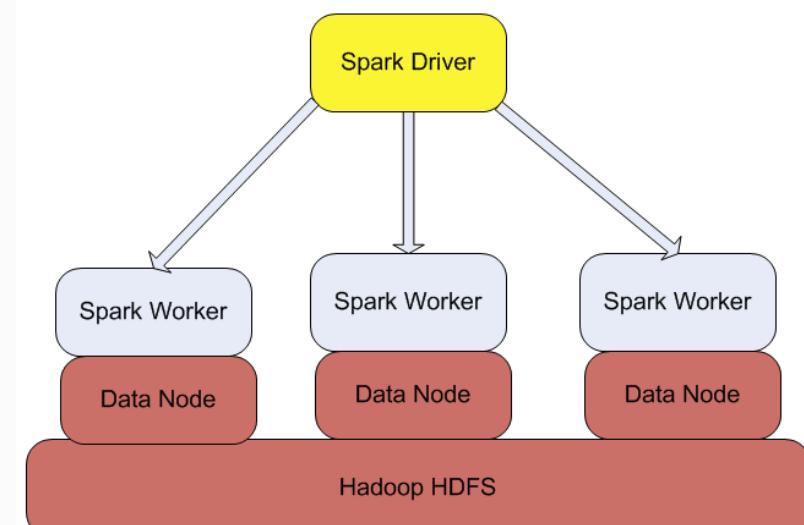
## Les « bancs de test »

- Données: nombreux catalogues (SDSS, 2MASS, etc.)
  - Ordre de grandeur jusqu'à ~ 60 Go et plusieurs dizaines de millions d'éléments en sortie (exemples: 2MASS 58Go, SDSS DR9 54Go, ~49 10<sup>6</sup> éléments en sortie)
- Ressources internes
  - Jusqu'à 6 nœuds (4 cœurs, 16Go, 1 To), des machines de bureau sous Ubuntu 14.04
- Ressources externes louées ponctuellement
  - 12 nœuds OVH (serveurs dédiés), 4 cœurs, 32Go, Raid 2\*2To, sous Ubuntu 14.04

Serveur de X-Match (12 coeurs, 32Go, 12To (15k tours))

## □ Les « bancs de test » (2)

- Architecture classique en utilisant directement les distributions d'Apache (Spark 1.5.0 pour Hadoop 2.6) + Java
- Mode standalone dans lequel Spark a son propre gestionnaire de cluster
  - Sans Apache Yarn, Mesos, ...
  - Ajout rapide de nouveaux nœuds



Crédits : BigHadoop



## □ Phase d'étude – Préparation des données

- Avant l'exécution les fichiers d'entrée sont stockés dans HDFS.
- Ces fichiers sont dans un premier temps chargés dans deux RDDs ((Resilient Distributed Dataset, une collection distribuée de données) simples où chaque ligne du RDD est un élément contenant des informations sur un objet dans le ciel.
- Chaque RDD est ensuite transformé en PairRDD (RDD contenant une paire de clé/valeur): à chaque élément du RDD est attribuée une clé représentant le numéro de pixel de la source grâce au découpage HEALPix du ciel.
- Les éléments des PairRDDs sont alors des couples (clé, valeur) où la valeur contient toutes les informations dont les coordonnées (ra, dec) de la source dans le système de coordonnées équatoriales.



## Phase d'étude – Préparation des données (2)

- Les PairRDDs sont ensuite distribués sur les différents nœuds du cluster.
- Cette distribution est faite sur la base d'un partitionnement par hachage où les PairRDDs sont découpés en partitions qui vont être stockés sur les nœuds.
- Le partitionnement par hachage consiste à regrouper tous les éléments ayant la même clé (même numéro de pixel) dans une même partition.
- Les partitions sont donc stockées sur des nœuds différents
  - Les éléments de même clé se retrouvent sur les mêmes nœuds
  - Cette distribution des données est essentielle pour la deuxième partie du programme.
- Enfin les PairRDDs sont enregistrés dans HDFS sous forme de fichiers binaires grâce à une méthode permettant de garder la structure (clé, valeur).



## Phase d'étude - Jointure

- Les fichiers binaires enregistrés précédemment sont directement chargés dans deux PairRDDs.
- Sur le deuxième PairRDD est appliquée une méthode qui duplique certaines sources dans les pixels voisins.
- Les deux PairRDDs sont ensuite joints au niveau de la clé. La jointure donne lieu à un nouveau PairRDD où les éléments sont de type (clé, valeur1, valeur2).
- La jointure étant faite sur la clé (numéro de cellule), deux sources proches peuvent être dans des cellules différentes et ne sont donc pas jointes (d'où la duplication des sources dans les cellules voisines pour limiter les effets de bord).

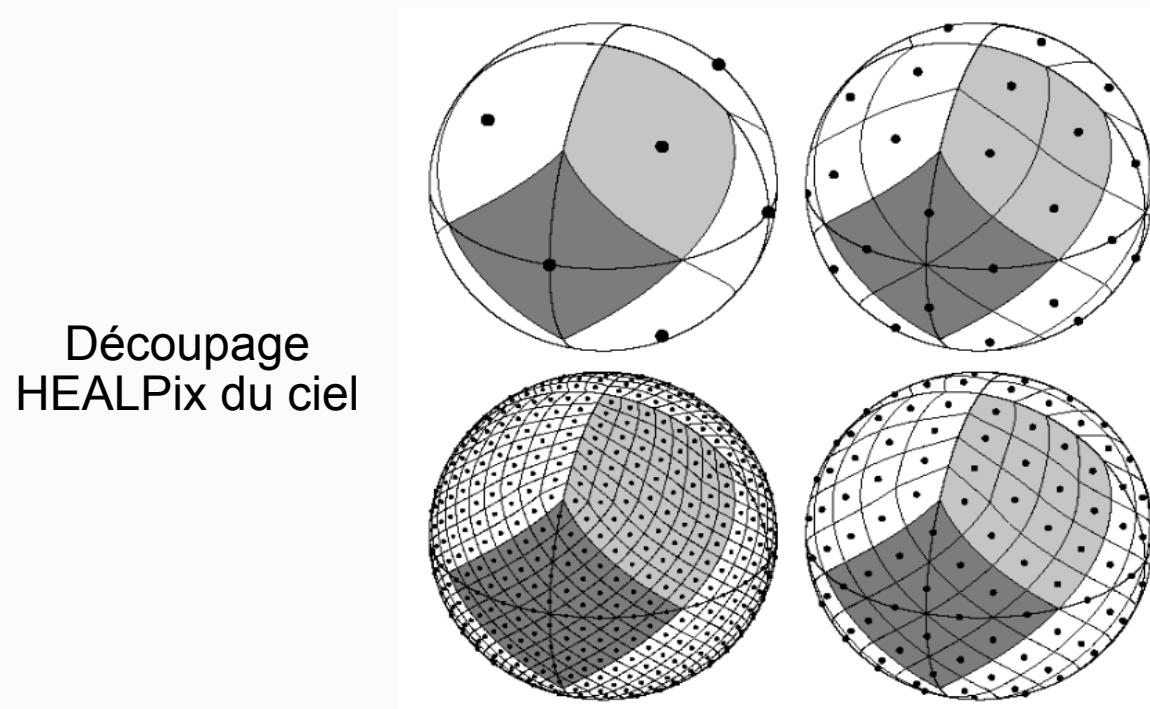


## Phase d'étude – Jointure (2)

- La duplication est effectuée de la manière suivante
  - Un cercle de rayon fixé est tracé autour de la source
  - Si des pixels voisins se trouvent en partie à l'intérieur de ce cercle, la source est alors dupliquée dans ces cellules voisines.
- Les éléments joints sont ensuite filtrés
  - Seuls les éléments joints dont la distance entre les deux sources est inférieure à un certain seuil sont gardés.
- Le résultat final est enregistré dans HDFS sous format texte pour une visualisation et une utilisation ultérieures.

## □ Illustration

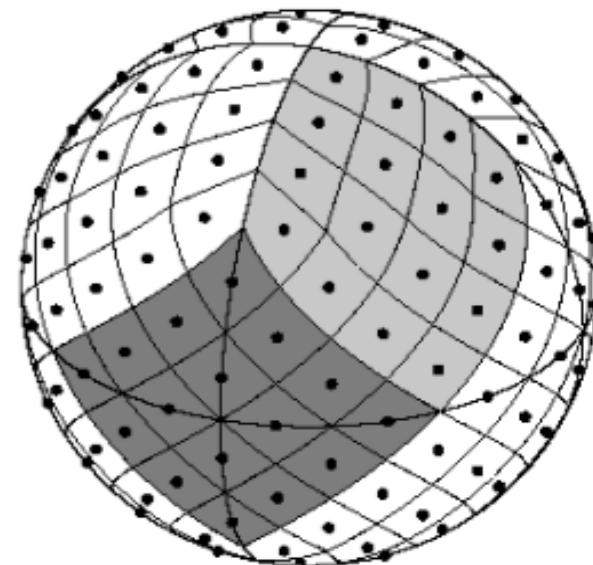
- Une implémentation du X-Match en MapReduce, Couples (clé = n°de pixel, valeur)





## Illustration (2)

- Effets de bord
  - Jointure floue
  - Duplication des sources dans les cellules voisines si besoin



Crédits : HEALPix – arXiv:astro-ph/0409513



## □ Phase d'étude – Co-location

- Lors du partitionnement par hachage des RDDs, les éléments de même clé sont placés sur les mêmes nœuds pour un RDD donné.
- Ceci n'implique pas que des clés communes à deux RDDs soient également sur les mêmes nœuds. Dans ce cas, cela engendre un temps de transfert des données entre les nœuds lors de la jointure, ce qui affecte les performances.



# Résultats

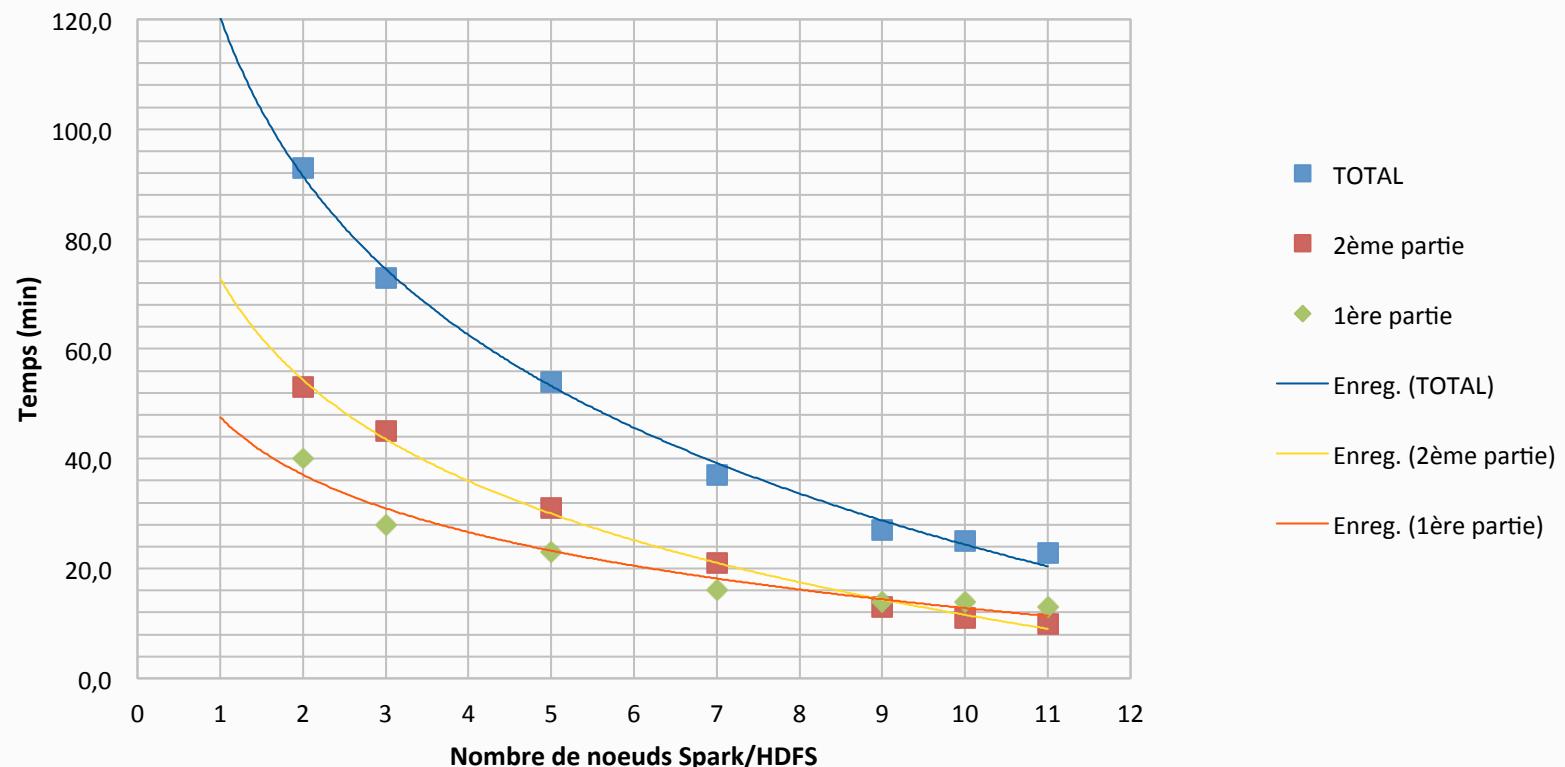
- Données en entrée (SDSS DR7 (sources primaires) et 2MASS): fichiers de 54GB et 58GB ; 357 175 411 et 470 992 970 d'objets
- Données en sortie: 49 208 820 d'éléments en sortie

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>		<b>40,0</b>	<b>28,0</b>	<b>23,0</b>		<b>16,0</b>		<b>14,0</b>	<b>14,0</b>	<b>13,0</b>	
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>		<b>53,0</b>	<b>45,0</b>	<b>31,0</b>		<b>21,0</b>		<b>13,0</b>	<b>11,0</b>	<b>9,9</b>	
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>		<b>93,0</b>	<b>73,0</b>	<b>54,0</b>		<b>37,0</b>		<b>27,0</b>	<b>25,0</b>	<b>22,9</b>	



# Résultats (2)

Temps de X-Match en fonction du nombre de noeuds



Le service de X-Match actuel nécessite 15 minutes de traitement pour ces mêmes données,  
ce qui correspond à la seconde partie (les données sont déjà préparée)



## Conclusion et perspectives

- Les résultats obtenus:
  - On obtient un temps inférieur à celui du service de X-Match
  - A partir de 8 nœuds cela peut devenir une alternative à l'architecture existante
  - En terme de coût l'ensemble de serveurs dédiés « en location » est intéressant (exemple: 8\*60\*12, env. 6000 euros / an)

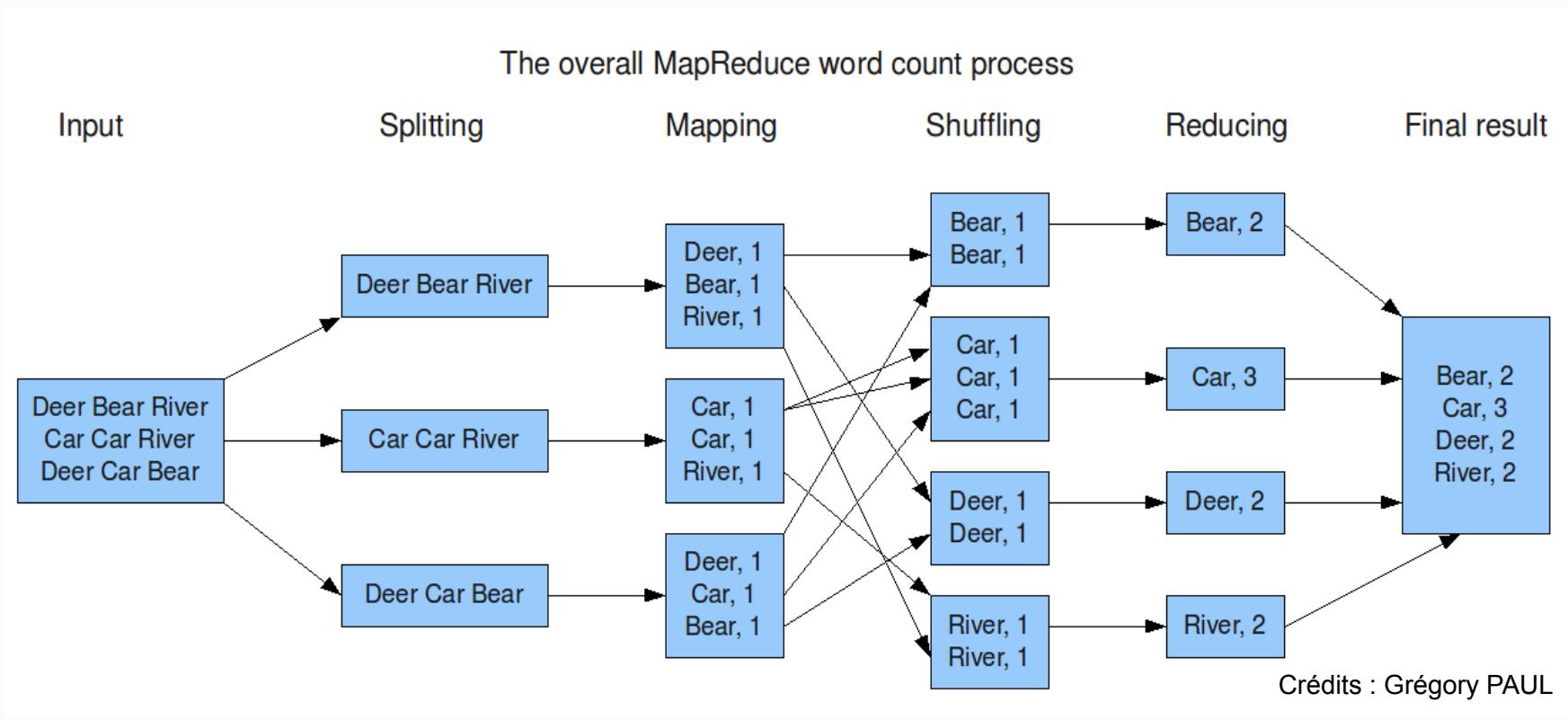


## □ Conclusion et perspectives (2)

- Goulot d'étranglement: « shuffle »
  - Optimisation possible (?) du code par la « co-location des données », pas de « block affinity groups » (nous sommes preneurs de bonnes idées pour avancer sur ce point...)
- Nous souhaitons réaliser de nouveaux tests avec plus de nœuds (explication)

## □ Phase de « shuffle »

- Redistribution sur les nœuds





# Liens

- 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/>