

Cas d'utilisation de Spark en Astrophysique: Cross Match de catalogues de sources

André Schaaff, François-Xavier Pineau

CDS, Centre de Données astronomiques de Strasbourg

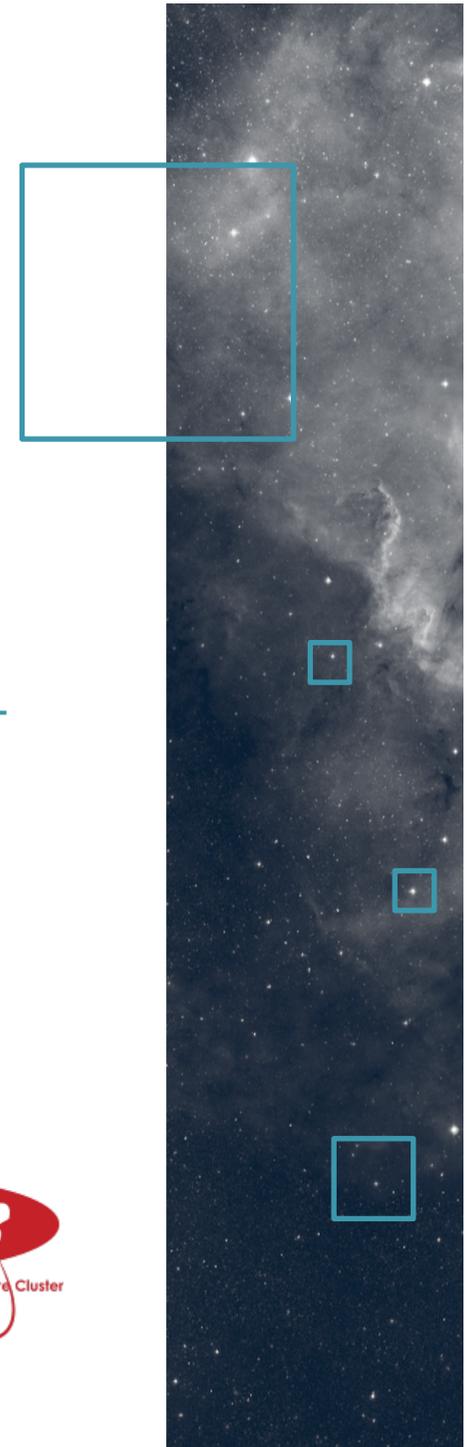
Noémie Wali, Paul Trehou

Elèves-ingénieurs UTBM, Université de technologie de Belfort-Montbéliard

Julien Nauroy

DI Paris-Sud

Journées Plateforme 6 & 7 octobre 2016, Clermont Ferrand



□ Le fil d'Ariane

Contexte

Apache Spark

La motivation

Les données et le service de « cross-match »

Les bancs de test

Première phase d'étude & Résultats

Deuxième phase (en cours)

Perspectives

□ Contexte

Une exploration continue de nouvelles technologies,
notamment « Big Data » (... et diversifiée)

Une première phase (9/2015 -> 2/2016 + prolongations)
« *Pour se faire la main, identifier les verrous, débiter des collaborations, etc.* »

Une deuxième phase en cours depuis début septembre
2016 pour 6 mois
« *Pour approfondir, introduire de nouvelles technologies (Docker)* »

□ Apache Spark

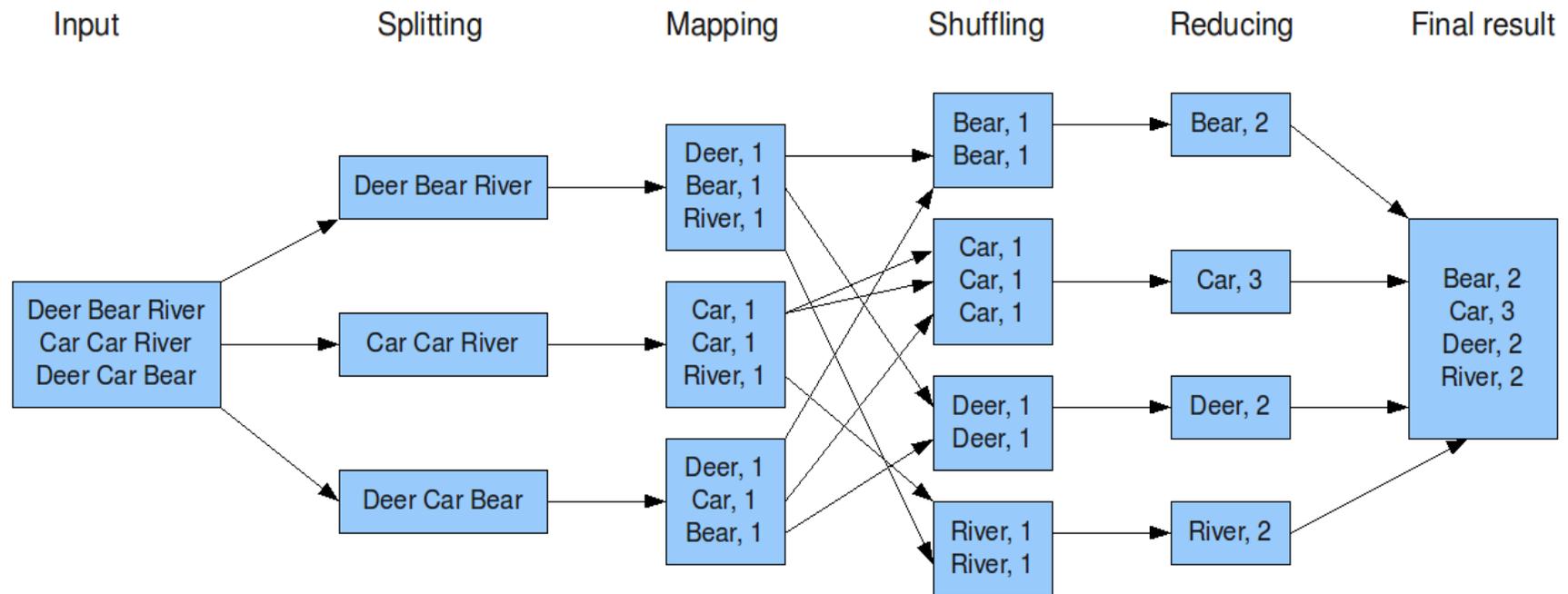
- 2009, premiers développements à l'AMPLab de l'UC de Berkeley (Matei Zaharia)...
- 2013, reprise par la fondation Apache
- 2014, V 1.0 avec pour particularité notable des performances élevées (par rapport à MapReduce)
- ... mi-2016 V2.0, unification des Datasets et des Dataframes

□ Apache Spark (2)

- Spark peut fonctionner avec divers systèmes de fichiers dont HDFS (Hadoop Distributed File System)
- Spark apporte des améliorations à MapReduce notamment au niveau du Shuffle (phase durant laquelle les données sont déplacées vers les noeuds sur lesquels s'effectuera la partie « reduce »)
- Il utilise un stockage en mémoire pour les étapes intermédiaires et réduit les accès disques -> très intéressant lorsque l'on travaille de façon itérative avec le même jeu de données
- Si la mémoire est insuffisante, Spark utilise l'espace disque

□ Illustration

The overall MapReduce word count process



Crédits : Grégory PAUL

□ Apache Spark (3)

- Spark introduit des modèles de données lui permettant de gérer une persistance des données durant les traitements
 - RDD (Resilient Distributed Dataset) pour stocker des objets (au sens OO), ils sont manipulables via des opérateurs ensemblistes type map ou reduce
 - Datasets pour représenter les données tabulaires manipulables via:
 - des commandes SQL
 - des fonctions SparkQL
 - des opérations ensemblistes
- API Spark: Scala, Java, Python, ...

□ La motivation (de cette étude)

- Nous souhaitons 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

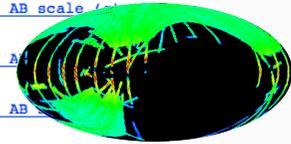
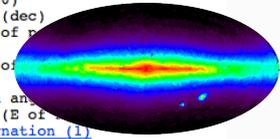
□ Les données...

- Données issues des catalogues de sources
- Exemples (nb sources):
 - 2MASS¹, 470,992,970
 - SDSS² DR9, 469,053,874

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

¹2MASS, Two Micron All Sky Survey,
²SDSS, Sloan Digital Sky Survey

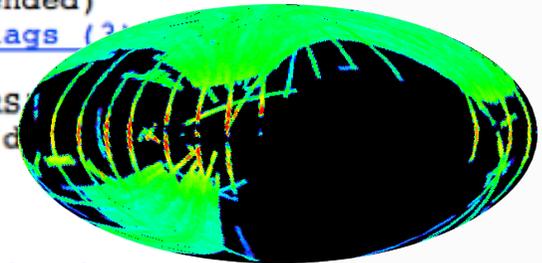
Bytes	Format	Units	Label	Explanations
1- 10	F10.6	deg	RAdeg	(ra) Right ascension (J2000)
12- 21	F10.6	deg	DEdeg	(dec) Declination (J2000) (dec)
23- 26	F4.2	arcsec	errMaj	(err_maj) Semi-major axis of error ellipse
28- 31	F4.2	arcsec	errMin	(err_min) Semi-minor axis of error ellipse
33- 35	I3	deg	errPA	[0,180] (err_ang) Position angle of 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	Jcmsig	?(j_cmsig) 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- 88	F6.3	mag	umag	?(u_m) u selected default magnitude (2)
90- 94	F5.3	mag	u/cmsig	?(u_cmsig) u default magnitude uncertainty (3)
96- 100	F5.3	mag	e_umag	?(u_msigcom) u total magnitude uncertainty (4)
102- 111	F10.1	---	usnr	?(u_snr) u Signal-to-noise ratio
113- 117	F6.3	mag	vmag	?(v_m) v selected default magnitude (2)
119- 123	F5.3	mag	v/cmsig	?(v_cmsig) v default magnitude uncertainty (3)
125- 129	F5.3	mag	e_vmag	?(v_msigcom) v total magnitude uncertainty (4)
131- 140	F10.1	---	vsnr	?(v_snr) v Signal-to-noise ratio
142- 146	F6.3	mag	zmag	?(z_m) z selected default magnitude (2)
148- 152	F5.3	mag	z/cmsig	?(z_cmsig) z default magnitude uncertainty (3)
154- 158	F5.3	mag	e_zmag	?(z_msigcom) z total magnitude uncertainty (4)
160- 169	F10.1	---	zsnr	?(z_snr) z Signal-to-noise ratio
171- 175	F6.3	mag	ymag	?(y_m) y selected default magnitude (2)
177- 181	F5.3	mag	y/cmsig	?(y_cmsig) y default magnitude uncertainty (3)
183- 187	F5.3	mag	e_ymag	?(y_msigcom) y total magnitude uncertainty (4)
189- 198	F10.1	---	ysnr	?(y_snr) y Signal-to-noise ratio
200- 204	F6.3	mag	mmag	?(m_m) m selected default magnitude (2)
206- 210	F5.3	mag	m/cmsig	?(m_cmsig) m default magnitude uncertainty (3)
212- 216	F5.3	mag	e_mmag	?(m_msigcom) m total magnitude uncertainty (4)
218- 227	F10.1	---	msnr	?(m_snr) m Signal-to-noise ratio
229- 233	F6.3	mag	nmag	?(n_m) n selected default magnitude (2)
235- 239	F5.3	mag	n/cmsig	?(n_cmsig) n default magnitude uncertainty (3)
241- 245	F5.3	mag	e_nmag	?(n_msigcom) n total magnitude uncertainty (4)
247- 256	F10.1	---	nsnr	?(n_snr) n Signal-to-noise ratio
258- 262	F6.3	mag	omag	?(o_m) o selected default magnitude (2)
264- 268	F5.3	mag	o/cmsig	?(o_cmsig) o default magnitude uncertainty (3)
270- 274	F5.3	mag	e_omag	?(o_msigcom) o total magnitude uncertainty (4)
276- 285	F10.1	---	osnr	?(o_snr) o Signal-to-noise ratio



□ Les données... - focus

SDSS² DR9, 469,053,874 sources

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	---	cl	<u>Type (class) of object (3=galaxy, 6=star) (1)</u>
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 -] <u>SDSS object identifier (2)</u>
49- 67	I19	---	objID	<u>SDSS unique object identifier (2)</u>
70- 84	A15	---	Sp-ID	<u>Spectroscopic Plate-MJD-Fiber identifier (7)</u>
86-104	I19	---	SpObjID	<u>Pointer to the spectrum of object, or 0 (7)</u>
106-124	I19	---	parentID	Pointer to parent (if object deblended)
126-141	A16	---	flags	[0-9A-F] <u>Photo Object Attribute flags (3)</u>
143-150	A8	---	Status	[0-9A-F] <u>Hexadecimal status (4)</u>
153-162	F10.6	<u>deg</u>	RAdeg	Right Ascension of the object (ICRS)
163-172	F10.6	<u>deg</u>	DEdeg	Declination of the object (ICRS) (d)
174-178	F5.3	<u>arcsec</u>	e_RAdeg	Mean error on RAdeg (raErr)
180-184	F5.3	<u>arcsec</u>	e_DEdeg	Mean error on DEdeg (decErr)
186-194	F9.4	<u>yr</u>	ObsDate	Mean Observation date
196	I1	---	Q	[0/5] <u>Quality of the observation (0=unknown):</u> 1=bad 2=acceptable 3=good 4=missing 5=hole (6)
198-203	F6.3	<u>mag</u>	umag	? <u>Model magnitude in u filter, AB scale (u) (5)</u>
204	A1	---	---	[:]



Les données... (dans 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. \(470992970 rows\)](#)

Simple Constraint | List Of Constraints

Query by [Constraints](#) applied on Columns (Output Order: + -)

Show Sort Column Clear Constraint

Show	Sort Column	Constraint
<input checked="" type="checkbox"/>	RAJ2000	deg (ra) Right ascension (J2000)
<input checked="" type="checkbox"/>	DEJ2000	deg (dec) Declination (J2000)
<input type="checkbox"/>	errMaj	arcsec (err_maj) Semi-major axis
<input type="checkbox"/>	errMin	arcsec (err_min) Semi-minor axis
<input type="checkbox"/>	errPA	deg [0,180] (err_ang) Position angle
<input checked="" type="checkbox"/>	2MASS	(char) (designation) Source designation
<input checked="" type="checkbox"/>	Jmag	mag ⁽ⁿ⁾ (j_m) J selected default magnitude
<input type="checkbox"/>	Jcmsig	mag ⁽ⁿ⁾ (j_cmsig) J default magnitude
<input checked="" type="checkbox"/>	e_Jmag	mag ⁽ⁿ⁾ (j_msigcom) J total magnitude
<input type="checkbox"/>	Jsnr	mag ⁽ⁿ⁾ (j_snr) J Signal-to-noise ratio
<input checked="" type="checkbox"/>	Hmag	mag ⁽ⁿ⁾ (h_m) H selected default magnitude

Search Criteria

Keywords: II/246/out

Tables: II/246/out

Preferences

max: 50

HTML Table

All columns

Compute

Distance ρ

Position angle θ

Distance (x,y)

Galactic

J2000

B1950

Ecl. J2000

[default]

Sort by Distance

+ order

- order

Position in:

Sexagesimal

Decimal

Mirrors

CDS, France

VizieR Result Page

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

[II/246/out](#) 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)

start AladinLite

Full	r	RAJ2000	DEJ2000	RAJ2000	DEJ2000	2MASS	Jmag	e_mag	Hmag	e_mag	Kmag	e_mag	Qflg	Rflg	Bflg	Cflg	Xflg	Aflg
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	12.565	0.055	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

Exemple: 2MASS et une recherche autour d'Andromède

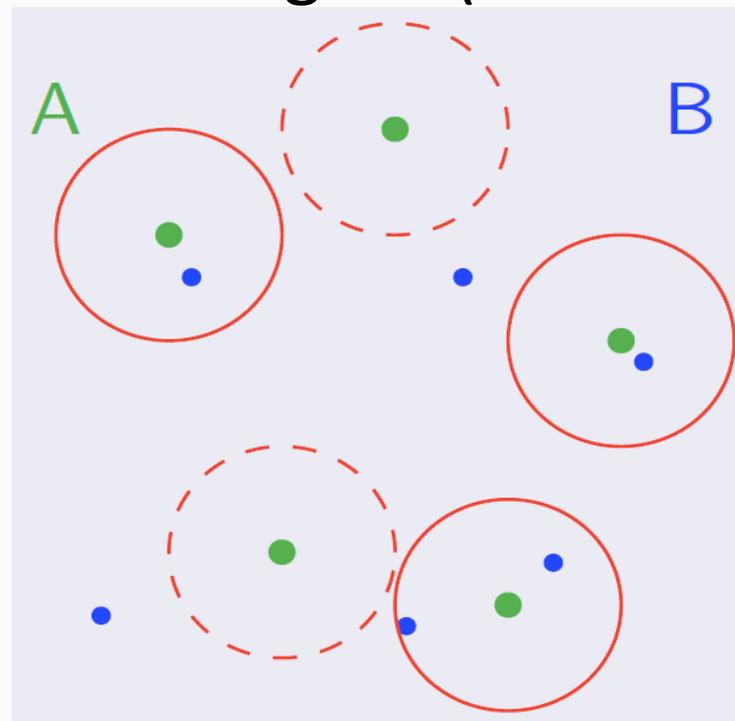
□ Les données... (dans VizieR) - focus

<i>Full</i>	<i>r</i>	<i>RAJ2000</i>	<i>DEJ2000</i>	<i>RAJ2000</i>	<i>DEJ2000</i>	<i>2MASS</i>	<i>Jmag</i>	<i>e_</i>	<i>P</i>
	arcmin	"h:m:s"	"d:m:s"	deg	deg		mag	mag	
△▽	△▽	△▽	△▽	△▽	△▽	△▽	△▽	△▽	△▽
<u>1</u>	0.0171	00 42 44.337	+41 16 08.53	010.684737	+41.269035	00424433+4116085	9.453	0.052	
<u>2</u>	0.0568	00 42 44.033	+41 16 06.91	010.683469	+41.268585	00424403+4116069	9.321		
<u>3</u>	0.0643	00 42 44.558	+41 16 10.38	010.685657	+41.269550	00424455+4116103	10.773	0.069	
<u>4</u>	0.0659	00 42 44.646	+41 16 09.21	010.686026	+41.269226	00424464+4116092	9.299		
<u>5</u>	0.0789	00 42 44.032	+41 16 10.83	010.683465	+41.269676	00424403+4116108	11.507	0.056	
<u>6</u>	0.0791	00 42 44.644	+41 16 10.67	010.686015	+41.269630	00424464+4116106	9.399		
<u>7</u>	0.1008	00 42 44.465	+41 16 01.65	010.685270	+41.267124	00424446+4116016	12.070	0.035	
<u>8</u>	0.1014	00 42 43.983	+41 16 02.84	010.683263	+41.267456	00424398+4116028	12.136	0.040	
<u>9</u>	0.1111	00 42 44.203	+41 16 00.99	010.684180	+41.266941	00424420+4116009	10.065		
<u>10</u>	0.1160	00 42 43.772	+41 16 04.53	010.682383	+41.267925	00424377+4116045	12.446	0.061	1
<u>11</u>	0.1194	00 42 43.866	+41 16 12.40	010.682777	+41.270111	00424386+4116123	9.977		1
<u>12</u>	0.1221	00 42 44.601	+41 16 14.16	010.685837	+41.270599	00424460+4116141	9.880		1
<u>13</u>	0.1288	00 42 44.147	+41 16 00.06	010.683944	+41.266682	00424414+4116000	12.565	0.055	

□ ...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

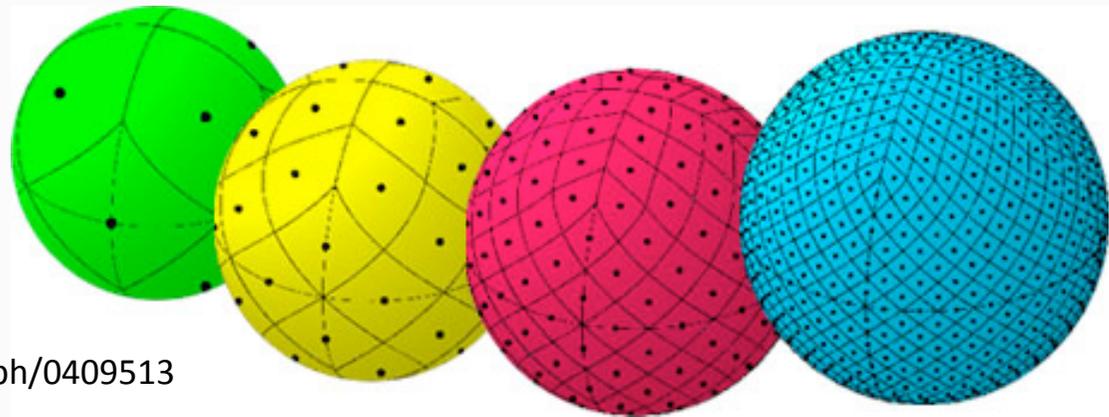


□ ...et le service de « cross-match » (2)

- 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).

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

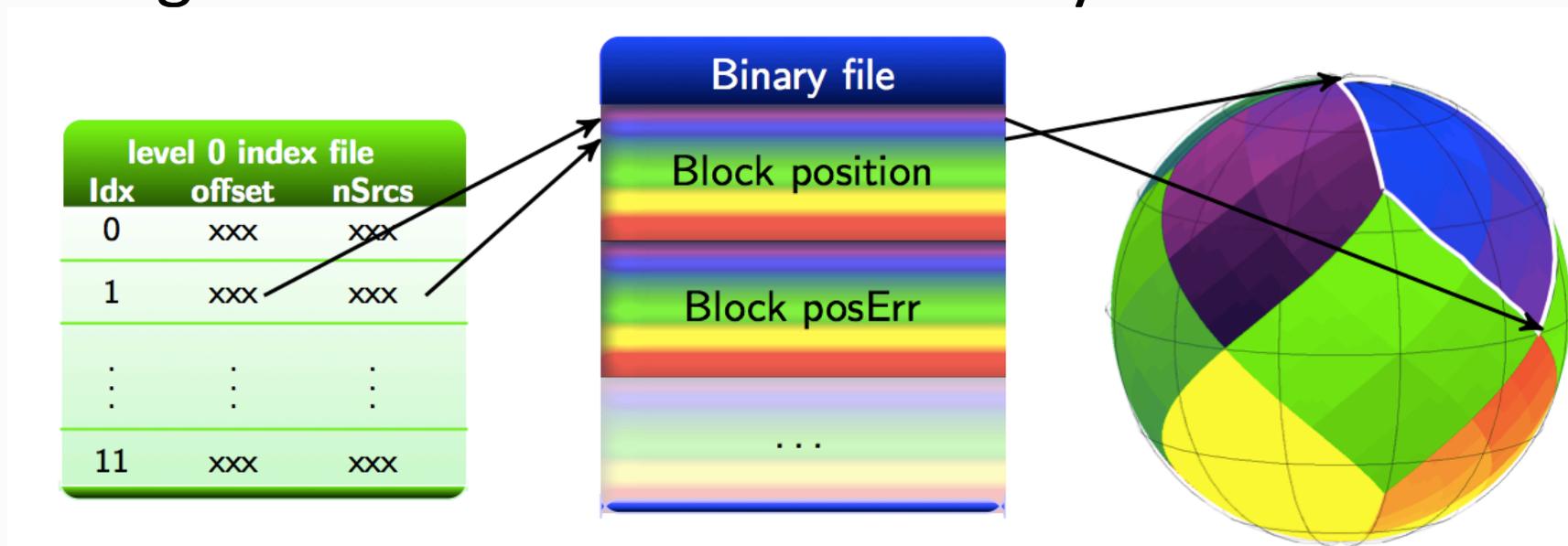
- Zone concernée
 - Tout le ciel: toutes les sources
 - Un cône: uniquement les sources à une certaine distance angulaire d'une position donnée
 - Une cellule HEALPix (pixellisation du ciel)



Crédits: HEALPix – arXiv:astro-ph/0409513

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

Exemple:
X-Match de
2MASS et 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 shows two selected tables: '2MASS' (470,992,970 rows) and 'SDSS DR9' (794,013,950 rows). Below the table selection, there are 'Cross-match criteria' and 'Cross-match area' sections. The 'Cross-match criteria' section has 'By position' selected with a radius of 5 arcsec. The 'Cross-match area' section has 'All sky' selected.

On the right, there's a 'Begin the X-Match' button and a 'Visualize and manage your cross-match jobs' section. This section contains a table of jobs:

Table 1	Table 2	Options	Begin	Status	Actions
2MASS	SDSS DR9	fixed radius	06/04/2016 at 10:21	executing	Abort

Below this, there's another 'Visualize and manage your cross-match jobs' section with a similar table, but the job status is 'completed':

Table 1	Table 2	Options	Begin	Status	Actions
2MASS	SDSS DR9	fixed radius	06/04/2016 at 10:21	completed	Get result, Download as CSV, Download as ASCII, Download as VOTable

At the bottom left, there's a date '6 & 7 octobre 20'.

□ Illustrations (2)

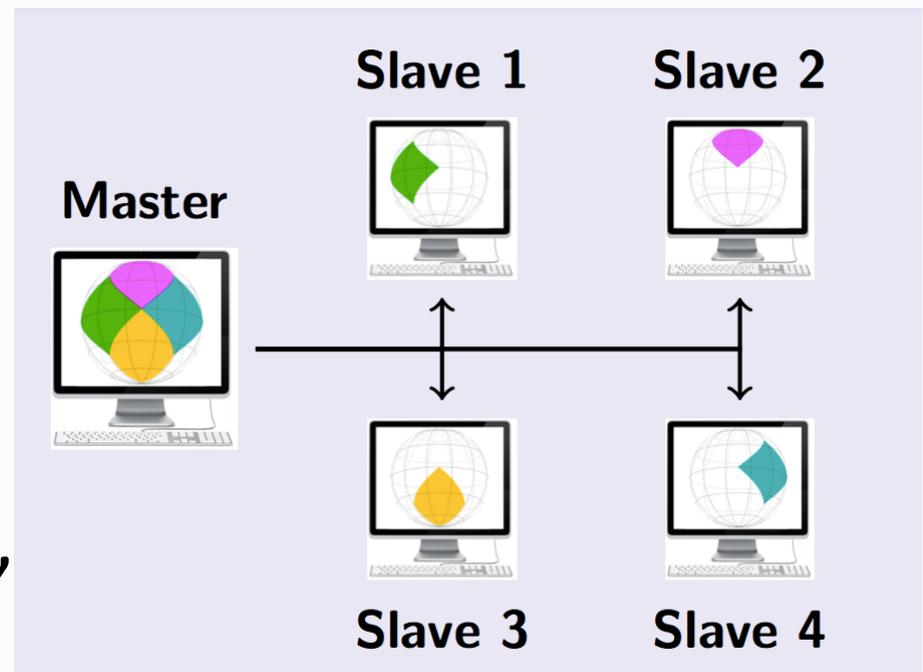
Exemple:
Un « extrait » du
résultat en CSV

```
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
```

```
angDist,2MASS,RAJ2000,DEJ2000,errHalfMaj,errHalfMin,errPosAng,Jmag,Hmag,Kmag,e_Jmag,e_Hmag,e_Kmag,e
MeasureJD,SDSS9,RAdeg,DEdeg,errHalfMaj,errHalfMin,errPosAng,umag,gmag,rmag,ima
imag,e_zmag,objID,cI,q_mode,flags,Q,ObsDate,pmRA,e_pmRA,pmDE,e_pmDE,SpObjID,zs
lass
0.305453,02595905+0000200,44.996055,+0.005565,0.170,0.160,76,16.376,15.770,15.
0,2451084.8062,J025959.06+000020.2,44.996116,+0.005624,0.002,0.002,90,19.548,1
28,0.006,0.007,0.007,0.013,1237663784217084122,6,1,0000201090020010,3,2003.885
```

□ 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 ?



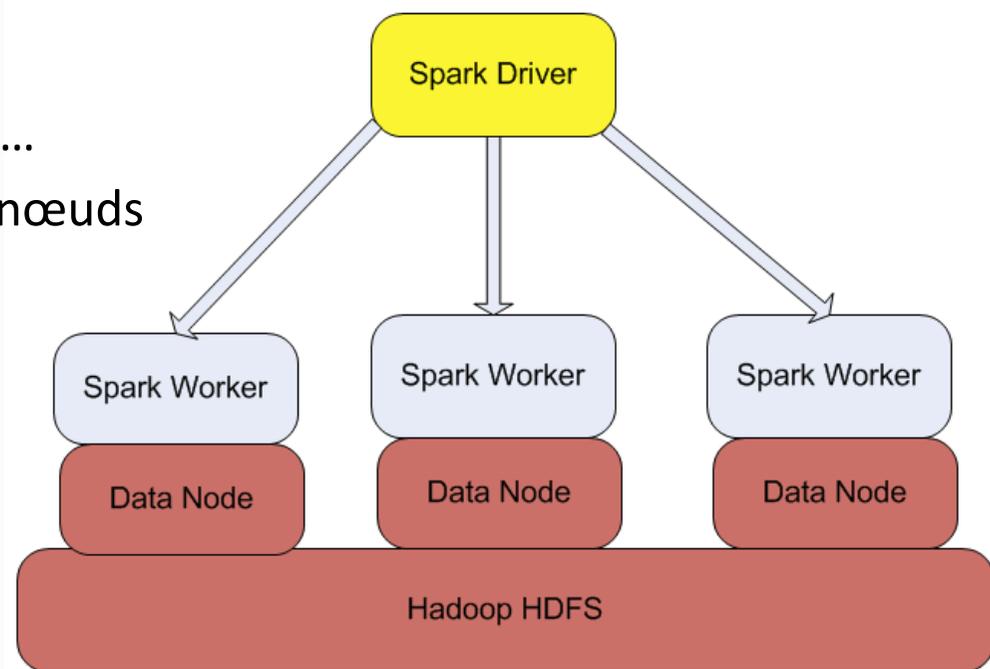
□ Les « bancs de test »

- 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, ~49 10⁶ 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 (2*6 coeurs, 32Go, 12To (15k tours))

□ Les « bancs de test » (2)

- Architecture classique en utilisant directement les distributions d'Apache (Spark 1.5* 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



* ...et maintenant Spark 2.0

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 **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 **jointes 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**).

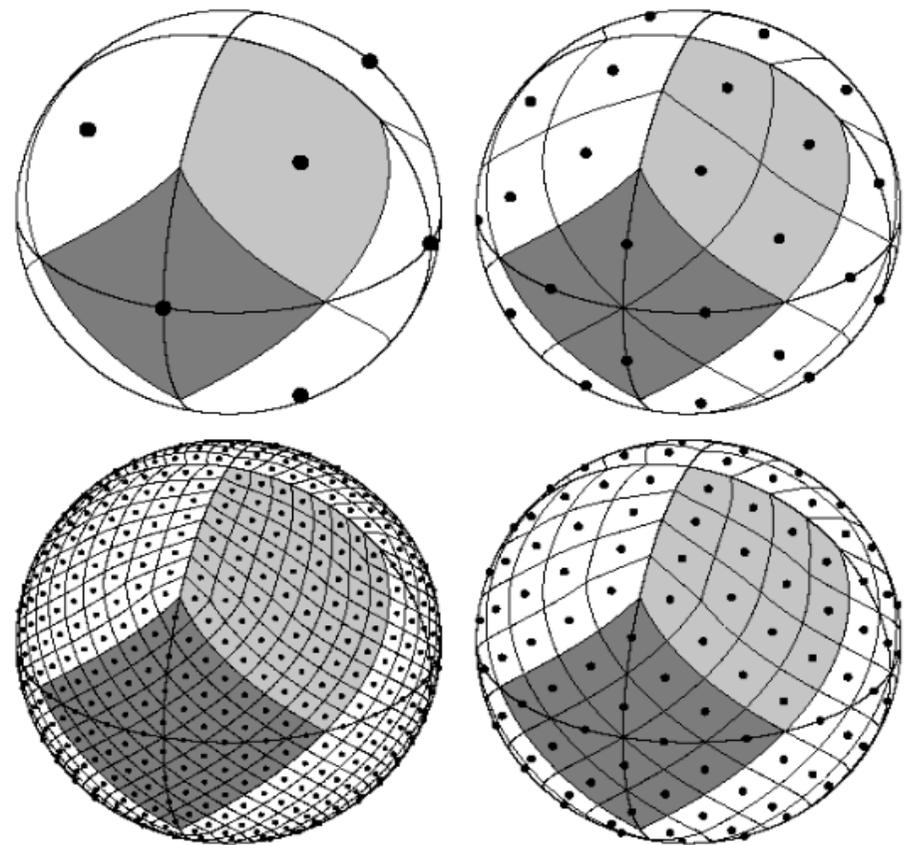
□ Phase d'étude – Jointure (2)

- **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

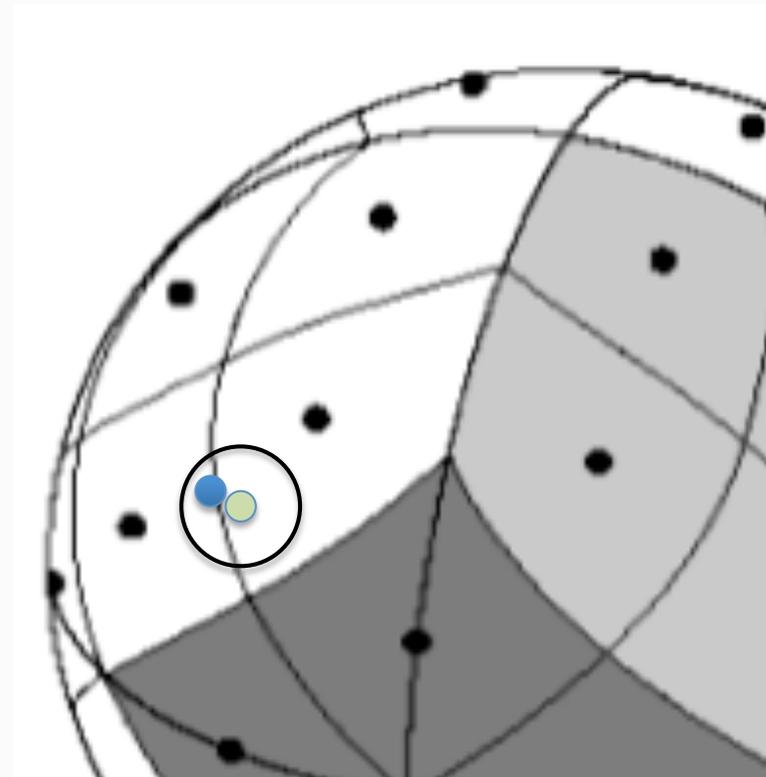
- Une implémentation du X-Match en MapReduce, Couples (clé = n°de pixel, valeur)

Découpage
HEALPix du ciel



□ 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

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
1ère partie : préparation des données		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
2ème partie : jointure		53,0	45,0		31,0		21,0		13,0	11,0	9,9
mapToPair (sdss7.bin)					7,2		4,7		3,3	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
TOTAL		93,0	73,0		54,0		37,0		27,0	25,0	22,9

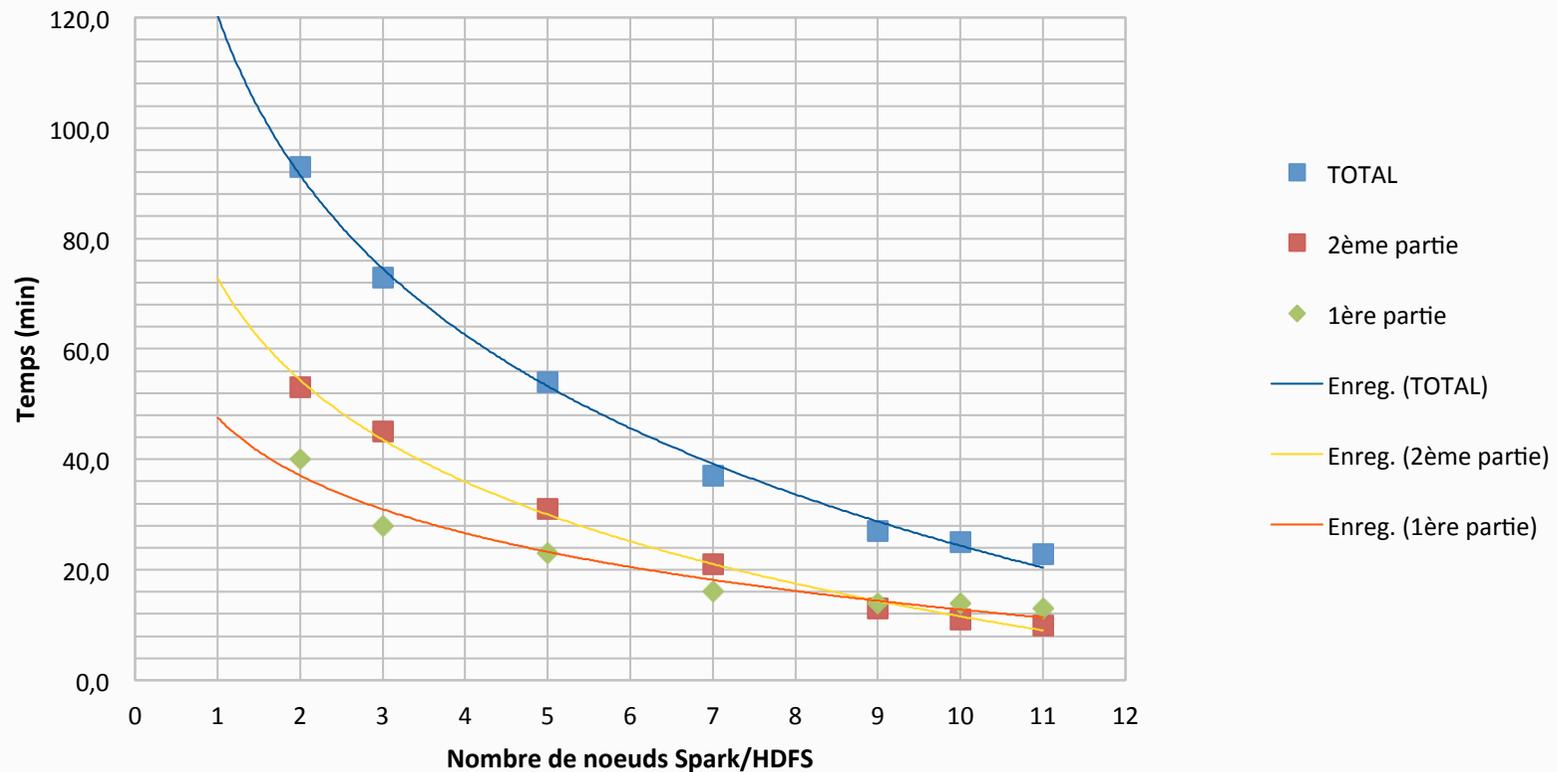
Le service de X-Match actuel nécessite **15 minutes** de traitement pour les mêmes données.

□ Résultats (2)- focus

7	8	9	10	11
16,0		14,0	14,0	13,0
4,9		4,9	4,8	4,7
2,7		2,0	2,3	1,5
5,2		5,2	5,1	5,0
3,6		1,9	1,6	1,4
21,0		13,0	11,0	9,9
4,7		3,5	3,0	2,9
8,3		5,5	4,9	4,3
7,6		3,4	2,4	2,3
37,0		27,0	25,0	22,9

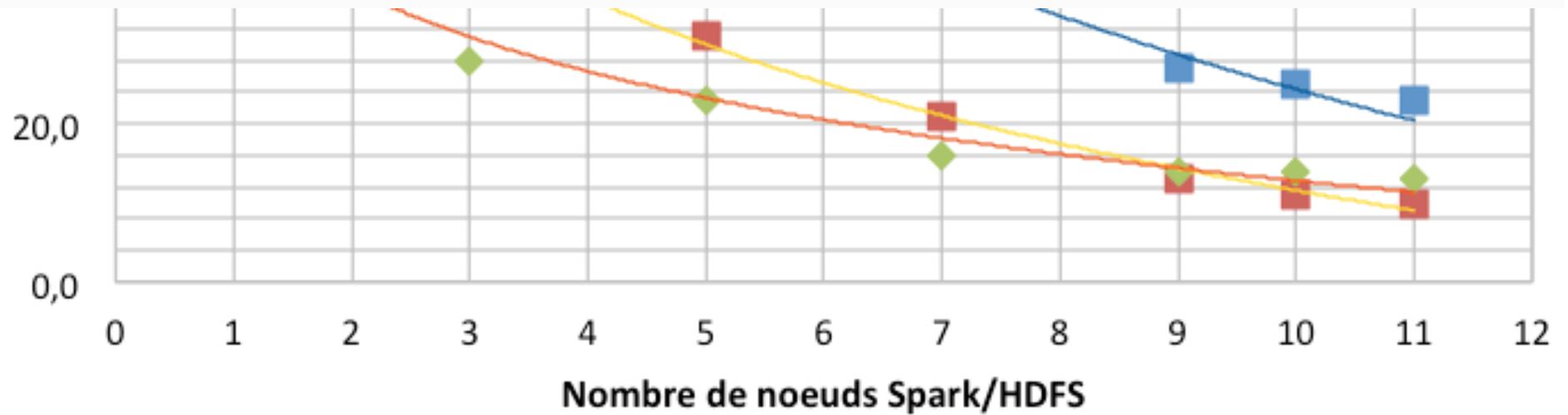
□ Résultats (3)

Temps de X-Match en fonction du nombre de noeuds



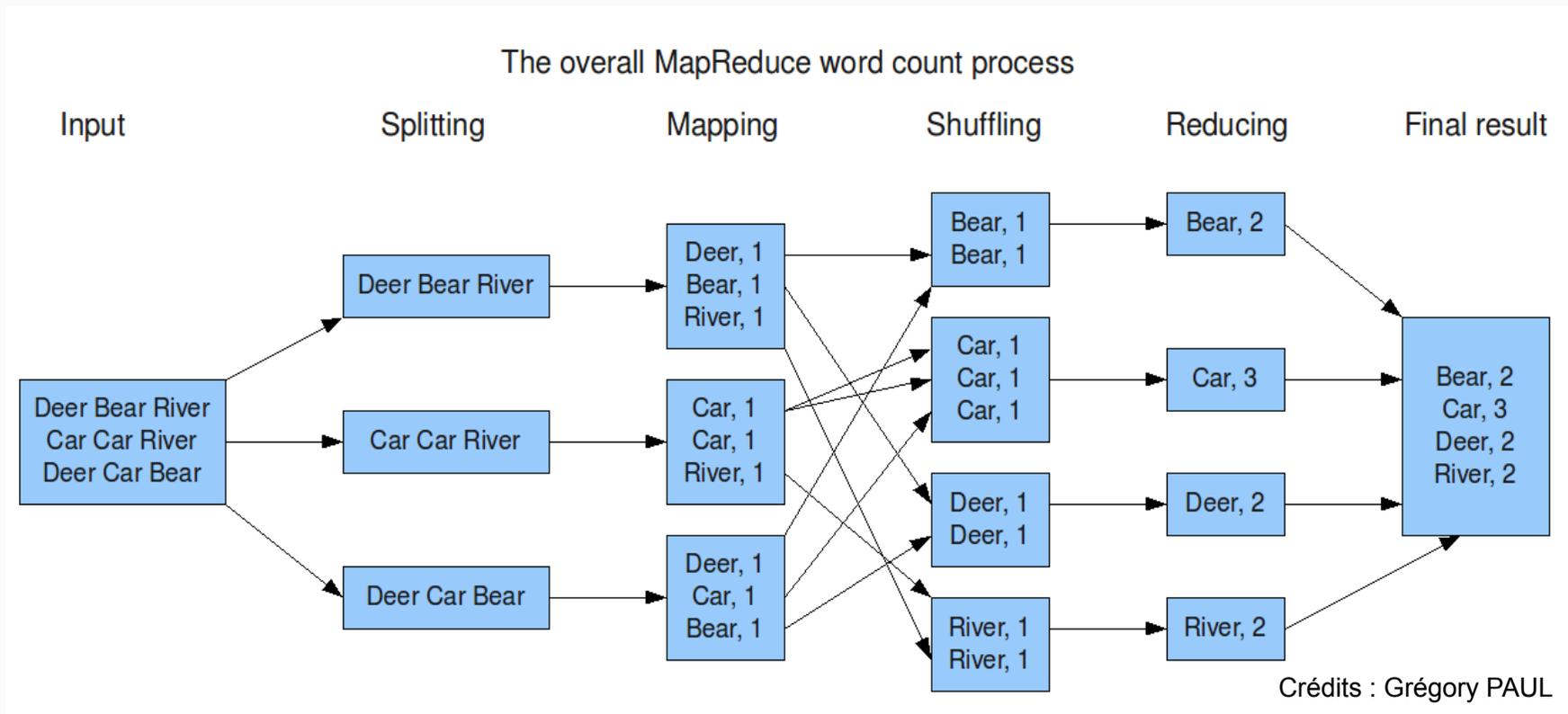
Le service de X-Match actuel nécessite 15 minutes de traitement pour les mêmes données (découpées en de multiples fichiers dans HDFS), ce qui correspond à la seconde partie (les données sont déjà préparées)

□ Résultats (4) - focus



□ Résultats (5) - Phase de « shuffle »

- Redistribution sur les nœuds



□ Résultats (6)

- Les résultats obtenus:
 - On obtient un temps inférieur à celui du service de X-Match
 - A partir de 8/9 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)
- N.B.: le banc de test n'était pas « optimisé »

□ Résultats (7)

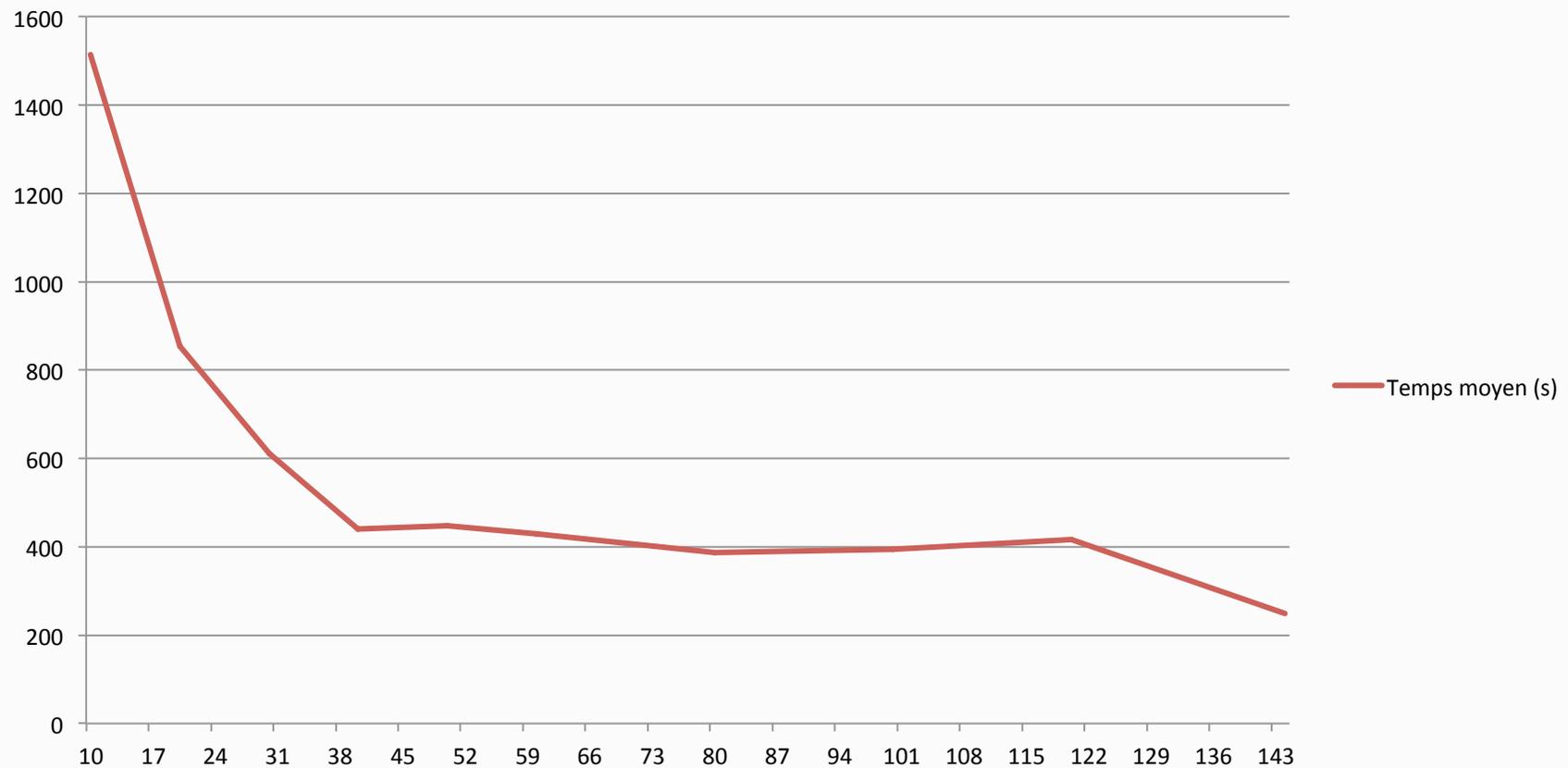
- Goulot d'étranglement: « shuffle »
 - Optimisation possible (?) du code par la « co-location des données », pas de « block affinity groups »

□ Prolongations de la phase 1

- Suite à la réunion LoOPS consacrée à Spark en avril 2016 au LAL, Julien Nauroy (DI Paris Sud) a réalisé des tests du X-Match avec son cluster Spark

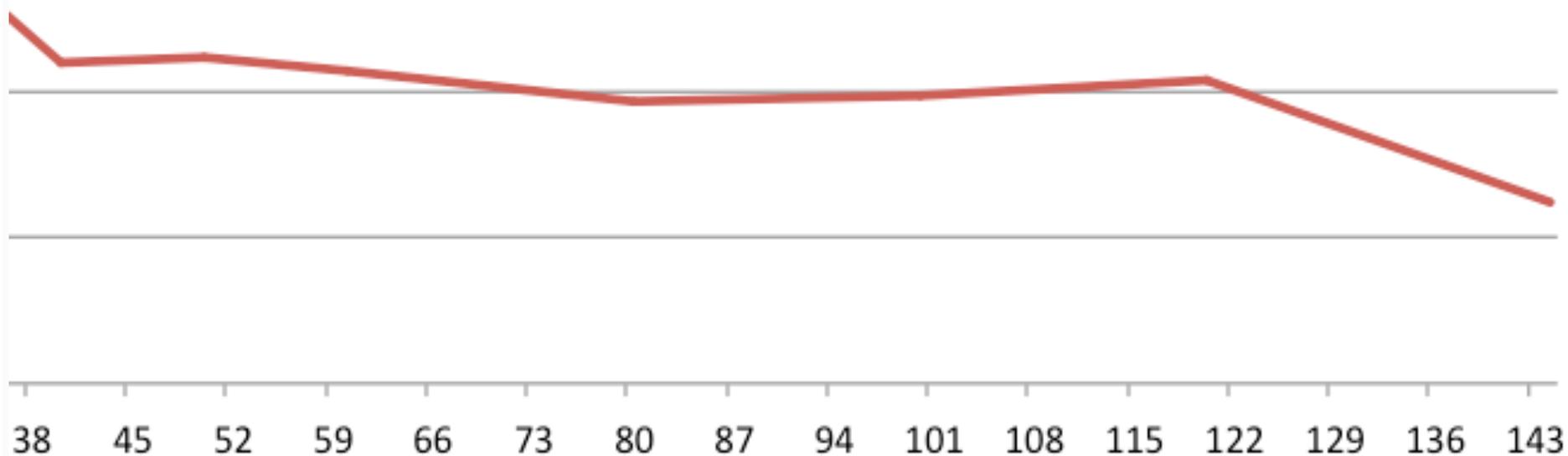
□ Résultats – benchmark J. Nauroy

Temps de calcul (s)



Credits : Julien Nauroy

□ Résultats – benchmark - focus



□ Résultats – benchmark J. Nauroy (2)

- Jusqu'à 40 cœurs, comportement linéaire
- Après 40 cœurs: plus d'amélioration
 - Goulot d'étranglement = débits disque/réseau
- Vers 144 cœurs...
 - Probablement dû à l'absence de « spill »

□ Résultats – benchmark J. Nauroy (3)

- Pistes d'optimisation
 - Lecture avec `session.read.csv`
 - Faire varier les partitions
 - des jeux en entrée
 - des structures intermédiaires
 - en particulier `spark.sql.shuffle.partitions`
 - Optimiser la taille mémoire

□ Résultats – benchmark J. Nauroy (3)

- Jointure de grands jeux pas optimale
 - 200 partitions par défaut
- Lecture CSV tout à fait viable
 - Mieux si on définit le schéma
- Beaucoup à gagner par essais successifs
 - Partitionnement
 - Types de données (Datasets, RDD)

□ Deuxième phase

- Débutée en septembre pour 6 mois
- Capitaliser sur le travail précédent, utiliser Spark 2.0
- Passage à Scala (l'API Java est « expérimentale »)
- Se rapprocher d'une mise en production par de nouveaux tests hardware
- Explorer les possibilités d'autres technologies en tant que complément, par exemple Docker

□ Deuxième phase (2)

- « Dockerisation » de Spark
 - Dans un contexte où l'on veut amener le traitement vers les données
 - Passer d'une simple exécution, par exemple d'un jar, à une exécution environnée
 - Intégration continue avec Drone / GitLab
 - Jupyter en cours d'évaluation
- Différents problèmes à traiter / résoudre notamment en terme de sécurité, de persistance des données, etc.
- Cette partie est plutôt en bonne voie

□ Perspectives

- Beaucoup d'avancées en seulement un mois, quelques ambitions pour les 5 mois restants
 - Améliorer de façon notable les performances au niveau du X-Match tout en restant dans le domaine du raisonnable (notion à définir !) en terme d'architecture et de coût
 - Amener la « dockerisation » dans un état stable et « en production » pour Spark mais également dans le cadre plus général du CDS

Crédits : Grégory PAUL



□ Et surtout...

- Echanges et collaborations !!!
- Nos différents travaux sont documentés et peuvent servir à d'autres

Crédits : Grégory PAUL

□ 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
- 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
- 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
- JavaSpark Api, <http://spark.apache.org/docs/latest/api/java/>
- HEALPix, <http://healpix.jpl.nasa.gov/>

Et Journée Spark au LoOPS, http://reseau-loops.github.io/journee_2016_04.html