



Cartographie de l'Univers en temps réel avec le framework Apache Spark

Julien Peloton, IJCLab/CNRS
29/06/2023

Cartographier l'Univers?

13.8 milliards d'années d'évolution

- À la frontière entre l'astronomie et la cosmologie.
- Un contenu hétérogène, et des techniques d'observation variées.
- Mesurer les distances?
- Encore beaucoup d'inconnus!

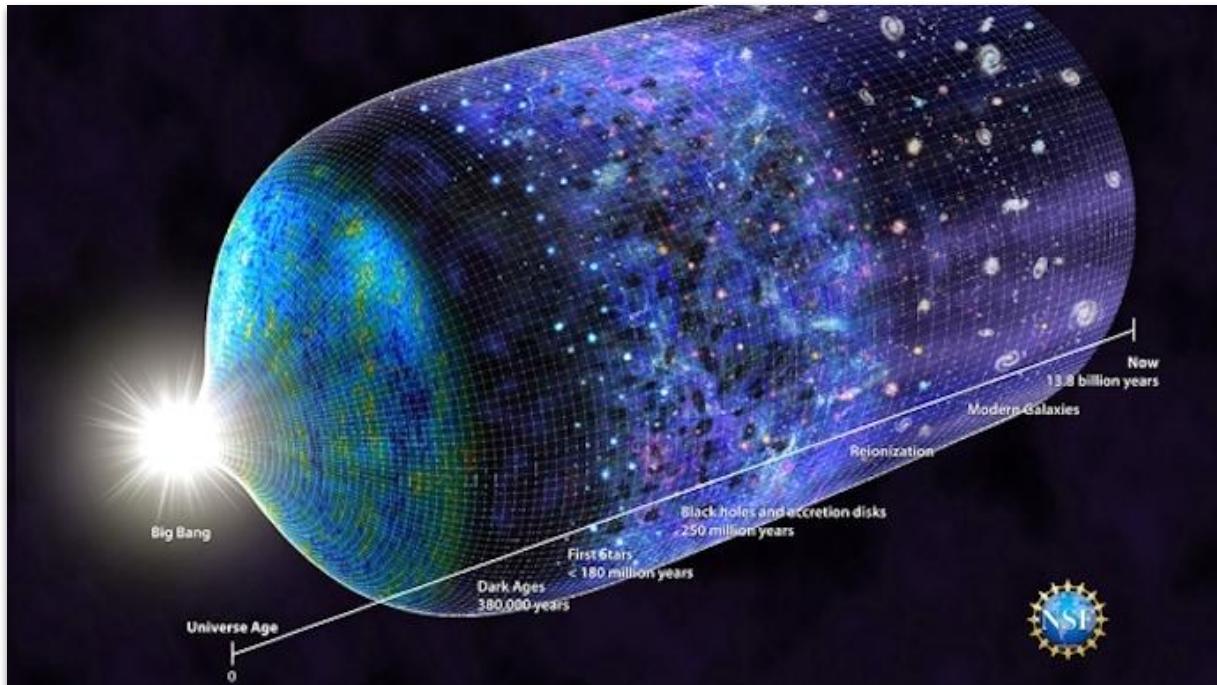




Photo Credit: Reidar Hahn, Fermilab



Photo Credit: Reidar Hahn, Fermilab

Challenge de données

L'observatoire Rubin (2024+) produira plus de 10 millions d'alertes par nuit, pendant 10 ans

- ~1 To/nuit: x50 au-dessus des flux existants

Fink: <https://fink-broker.org>

Prototypage sur le Zwicky Transient Facility (200,000 alertes/nuit, 20 Go/nuit)



Feuille de route

Le défi n'est pas tant de collecter ces données que des les interpréter.

- Comment mettre les scientifiques au coeur d'un projet big data ?
- Centraliser le calcul, decentraliser la science

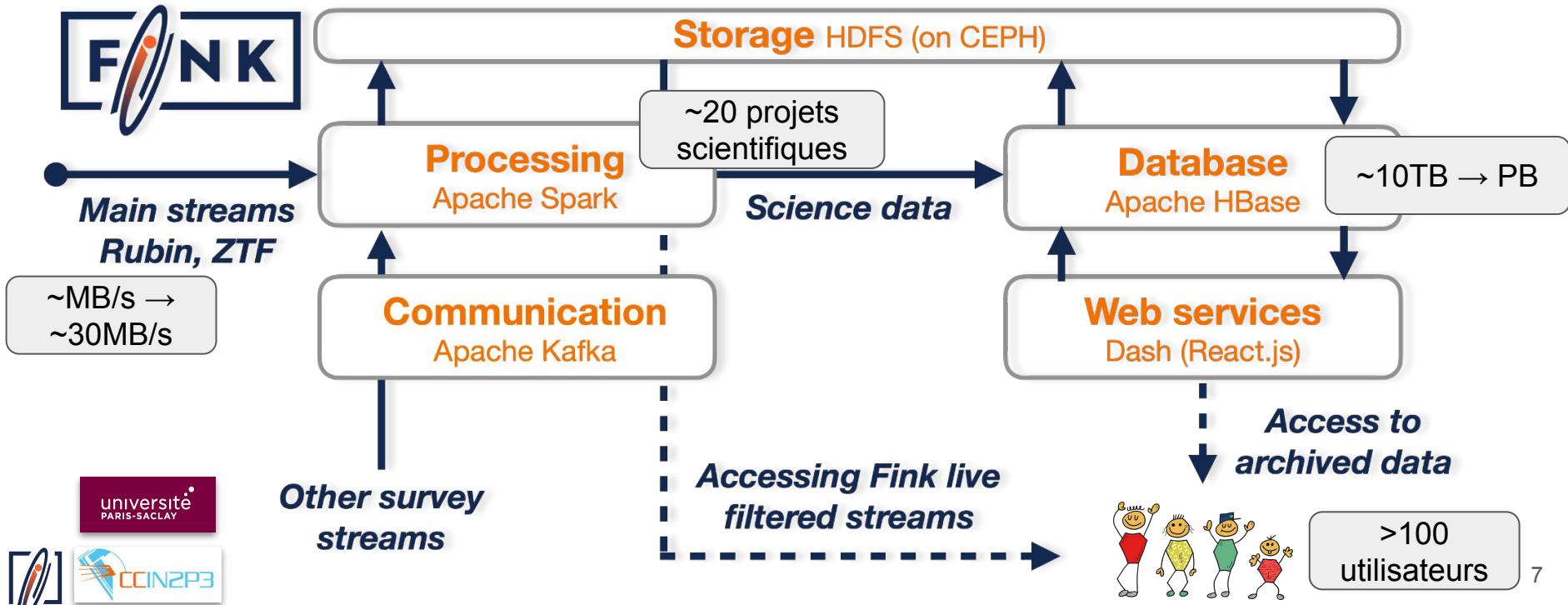
Idéalement :

- **[Traitement en flux]** Traiter les données en temps réel, et avertir les utilisateurs de cas intéressants.
- **[Héritage]** Permettre à tout utilisateur de modifier simplement le programme pour y incorporer sa logique scientifique.
 - Corollaire : être capable d'appliquer des programmes anciens sur de grosses masses de données – sans connaître les détails informatique
- **[Diffusion]** Stocker les données, et permettre une exploration *visuelle & programmatique*, quel que soit le volume sous-jacent.



Plateforme Fink

Cloud VirtualData (200 vCPU, 35TB stockage) & CC-IN2P3 (500 vCPU, 1PB stockage)



Services utilisateurs

Service utilisateurs

- Broker (*200,000 alertes par nuit – 7 To traité en flux depuis 2019*)
- Portail Web & API (*100 utilisateurs au quotidien – 3,000+ requêtes par jour*)
- Transfert de données massives (*~500 millions d'alerte téléchargées par mois*)
- Extension Apache Spark (spark-fits, spark3D)



Broker: de l'information à la science

Broker: 200,000 alertes/nuit – bientôt 10 millions alertes/nuit.

Les alerts contiennent de l'*information* – nous avons besoin d'experts en astronomie et cosmologie pour en extraire la *science*:

- Plus de 30 scientifiques contribuent au projet (international)

Utilisation d'UDF dans Apache Spark Structured Streaming pour

- de l'inférence (e.g. scikit-learn, pytorch, tensorflow)
- des requêtes distantes (e.g. centre de données Strasbourg)
- des jointures de flux de données.

Templates de code & jeux de données disponibles pour les utilisateurs, et plateforme notebook avec Apache Spark. L'API unifiée de Spark permet de passer du statique au streaming facilement !



Partie utilisateur

Syntaxe pour les entrée/sortie imposée,
mais traitement interne libre.

Exemple typique : chargement d'un
modèle en apprentissage automatique
puis inférence.

→ Depuis 2020 : découverte de +900
supernovae, une exoplanète, des
astéroïdes non-détectés, ...

Difficulté: profilage (Spark 3.4+), accès
données externes (broadcast).

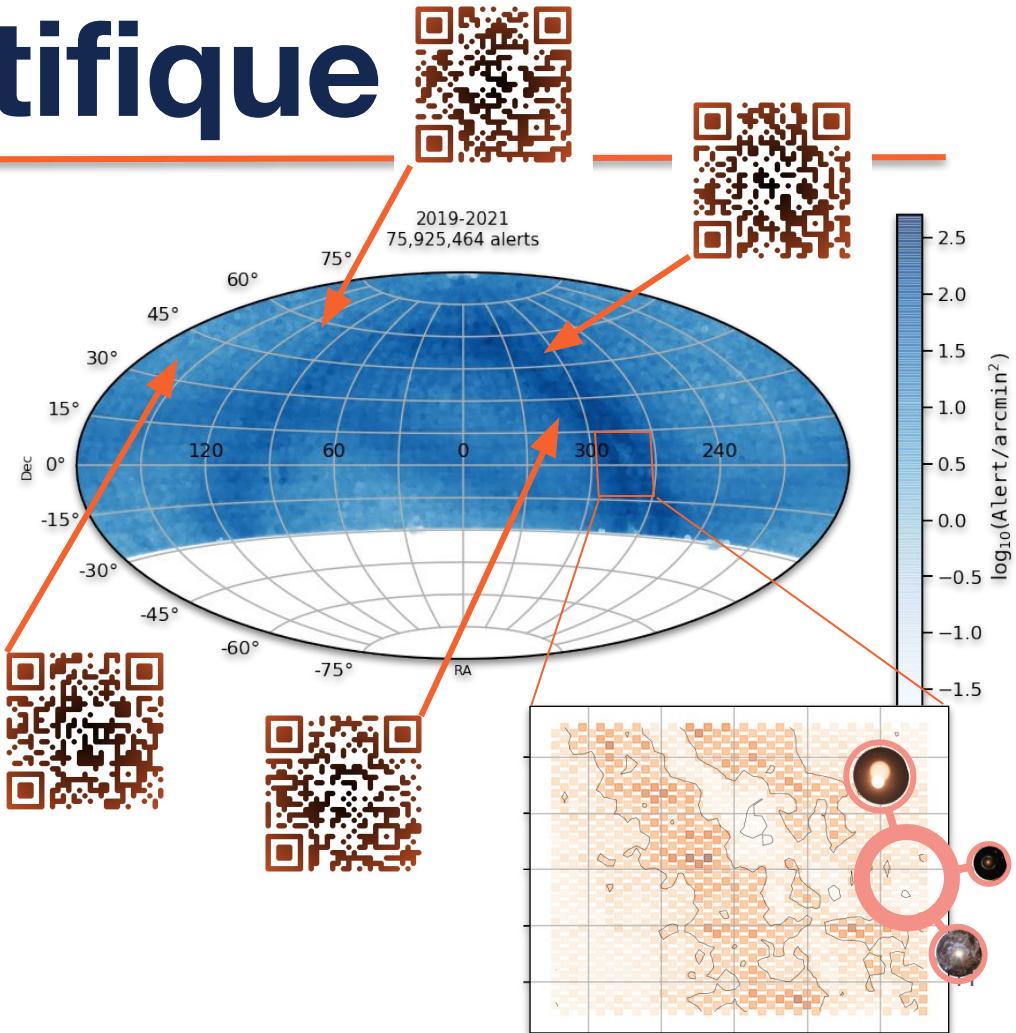
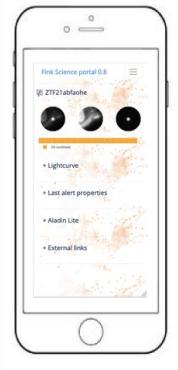
```
1 @pandas_udf(output_type, PandasUDFType.SCALAR)
2 def do_something(*inputs):
3     """
4     """
5     model = load(...)
6
7     output = model.predict(*inputs)
8
9     return pd.Series(output)
```



Portail Scientifique

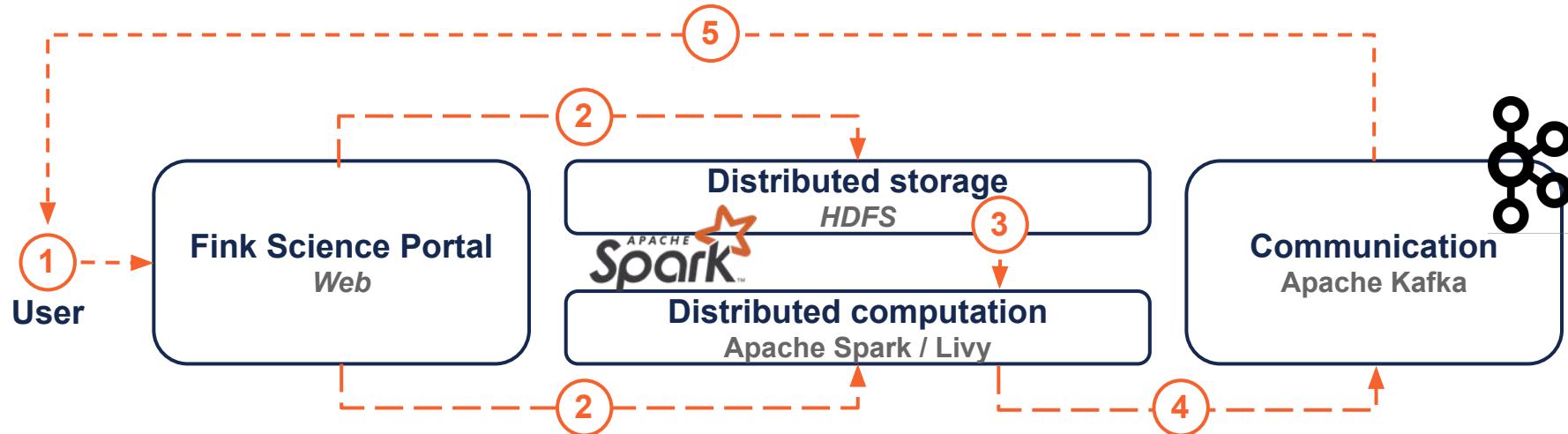
170 million alerts received since 2019:
<https://fink-portal.org/stats>

API REST disponible:
<https://fink-portal.org/api>



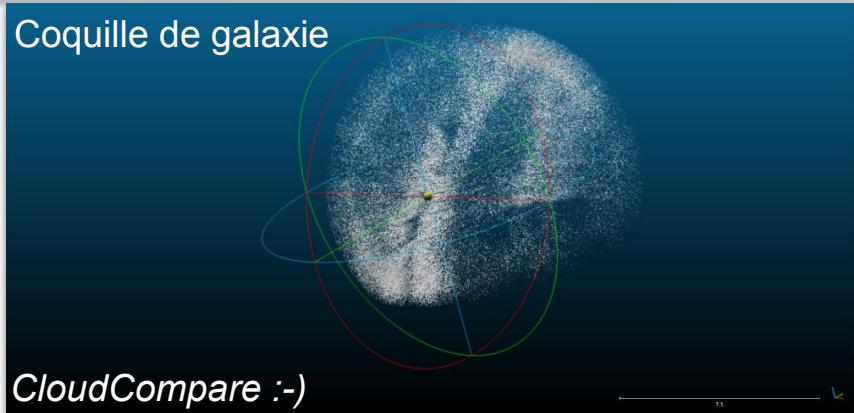
Transfert de données

Question: comment transférer des données massive d'un point A à un point B, de manière flexible (arrêt/stop/continue), et personnalisable (filtres) ?

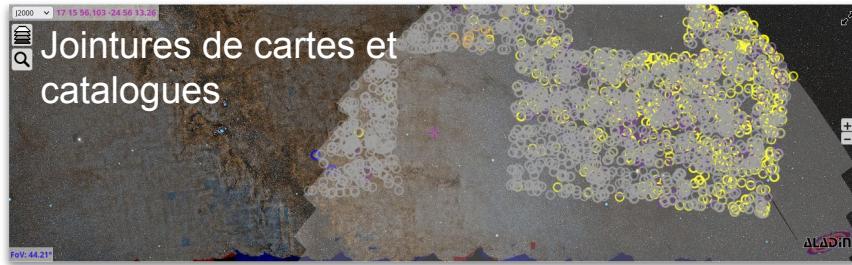


Quelques cartes de Fink

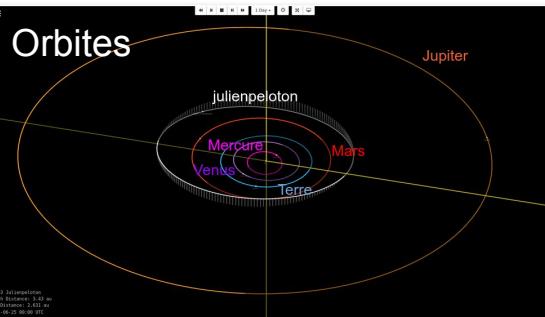
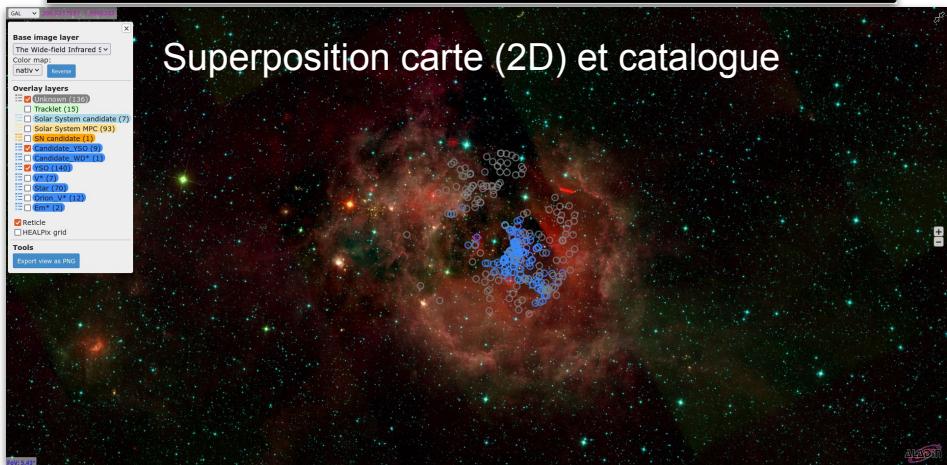
Coquille de galaxie



CloudCompare :-)



Superposition carte (2D) et catalogue



Outils spécifiques: spark-fits

Besoin de lire des formats de données
astro FITS: [spark-fits](#)

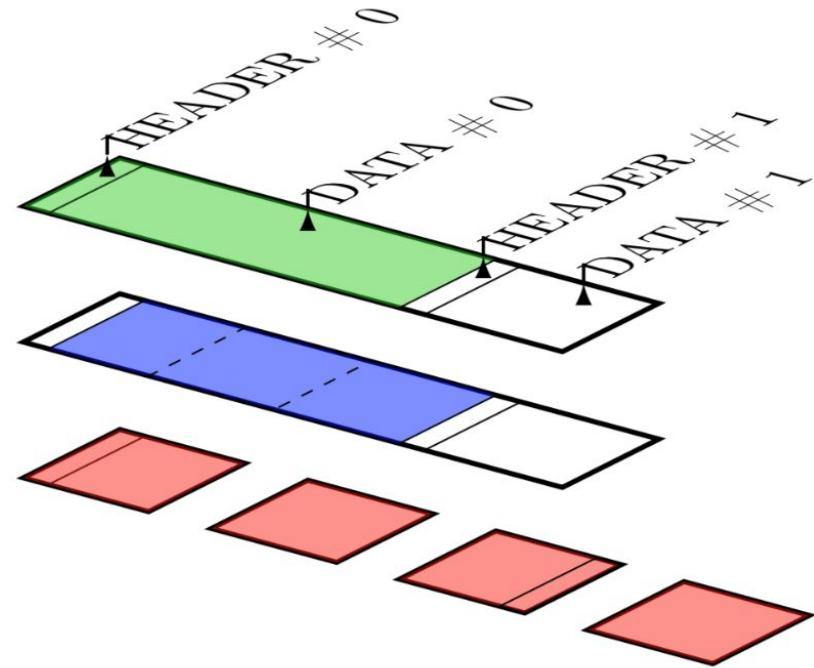
FITS data source pour Spark SQL.

- étend Hadoop *FileInputFormat* pour les images & tables.
- Testé jusqu'à 1 To

Publication:

<https://doi.org/10.1007/s41781-018-0014-z>

Performances (début I/O) comparables
aux autres connecteurs Spark.



Outils spécifiques: spark3D

Données 2D avec Spark:

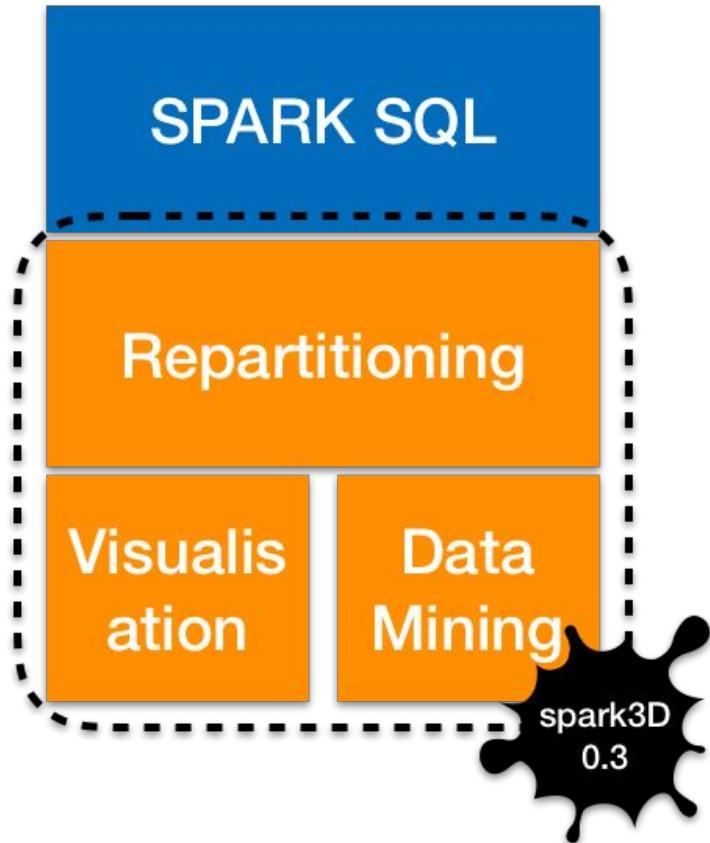
- Geotrellis, Magellan, Apache Sedona (Geospark), GeoMesa, ...

Très peu sur la 3D !

spark3D

- Etudiant via le Google Summer of Code 2018
- These de Master 2020

Partitionnement distribué en 3D,
requêtes spatiales 3D.



Quelques cas

Recherche de voisins (KNN):

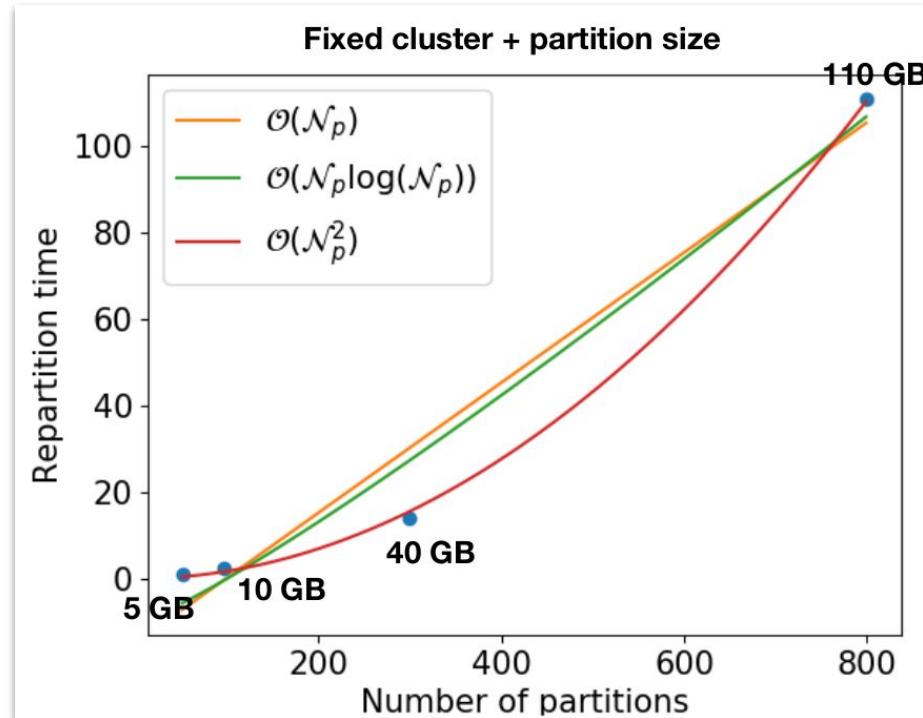
- 6 milliards de positions, K=1000
for un point unique en $O(10)$ sec.

Jointure de catalogues:

- 6 milliards x 190 millions de positions de galaxies en $O(10)$ sec.

Recherche de clusters (RP-DBSCAN
impl. by Song, H. and Lee, J.):

- 1.2 milliards de galaxies, centres trouvés en $O(10)$ min.

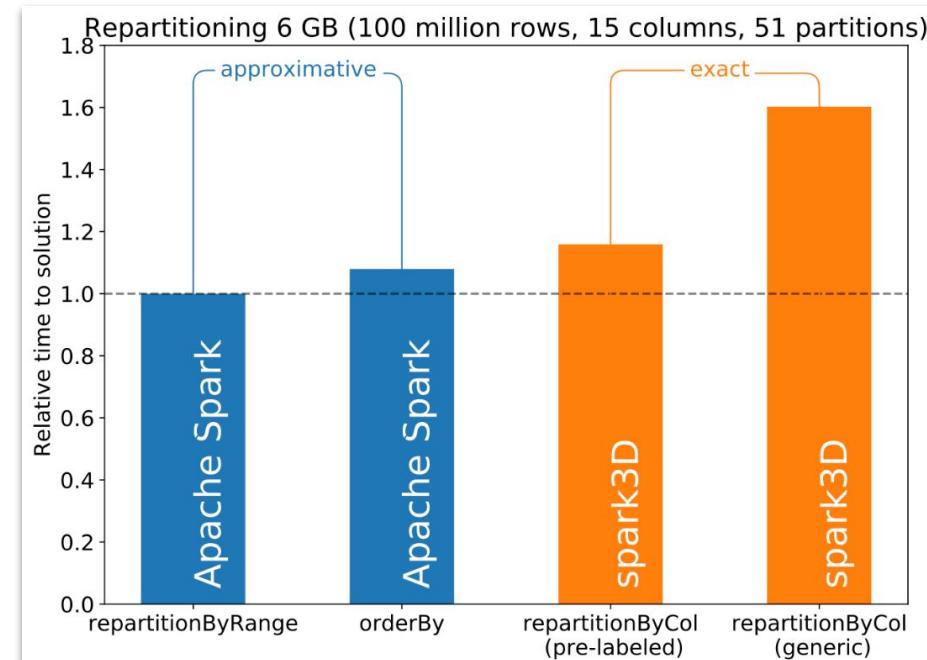


Repartitionnement exact

Le partitionnement de DataFrame sur Spark est non déterministe : échantillonage.

Balance entre performance et précision.

Spark3D expose des méthodes pour un partitionnement exact, sans trop de pénalité.



Conclusion

Le cloud est une plateforme adaptée pour le traitement de données massives.

Un des défis majeurs pour le big data est de faire le lien avec l'héritage de programmes existants et d'offrir des interfaces (ou services) intuitives

- Comment mettre les scientifiques au cœur d'un projet big data ?

Apache Spark apporte suffisamment de performance et de flexibilité pour l'instant dans nos analyses. D'autres acteurs ont émergés, e.g. Dask.

Cartographier l'Univers n'est pas fini :-)





Simulation cosmologique (LSST-DESC) / Crédit photo & visualisation : G. Barrand.