

Introduction à Spark

Jean-Paul LE
Geoffrey ALDEBERT

Définition de Spark

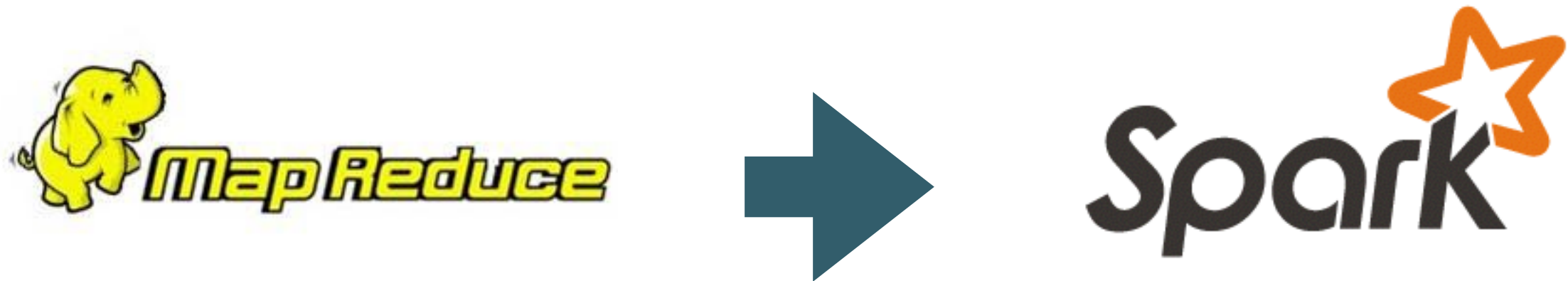
Définition de Spark

Généralité

- Framework de data processing pour le big data
 - Polyvalent : **API haut niveau** qui facilite les traitements sur Hadoop
 - Calcul en mémoire et parallélisé
 - Fault-tolerant
 - Codé en Scala

Définition de Spark

Généralité



- **Rapide :**
 - Traitement **en mémoire** :
 - Plus rapide que MapReduce
 - Evite les IO inutiles sur disques
 - Moteur d'**optimisation du DAG** (en map reduce, on optimise les 2 seules tâches map/reduce, alors que Spark optimise toute la chaîne)

Définition de Spark

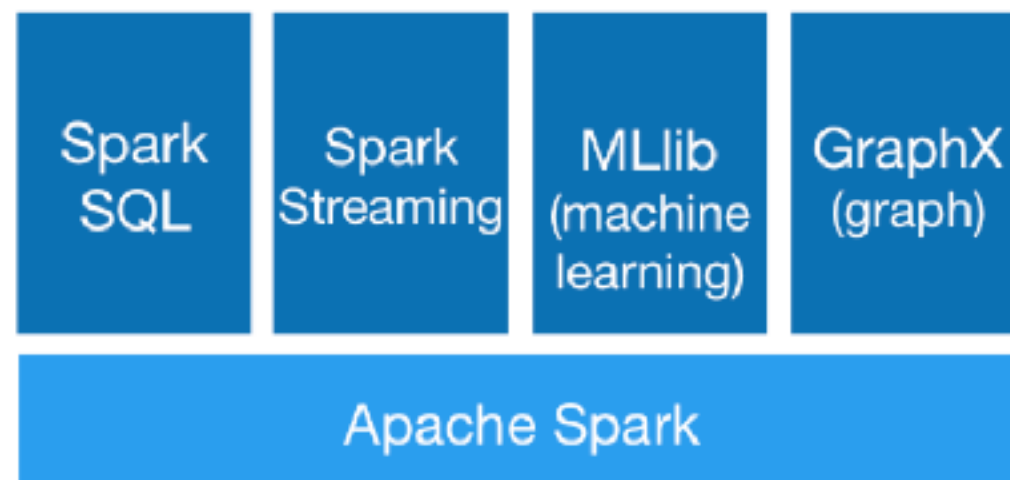
Généralité

- Les distributions de Spark :
 - HortonWorks
 - DataBricks
 - MapR
- Langages:
 - Scala
 - Java
 - Python
 - R
 - ...

Définition de Spark

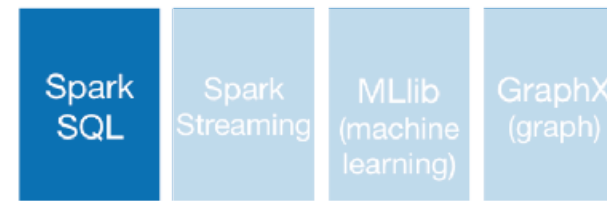
Généralité

- Framework polyvalent
 - Manipulation et streaming de données structurées
 - Algo de machine learning distribué
 - Manipulation de graphes



Définition de Spark

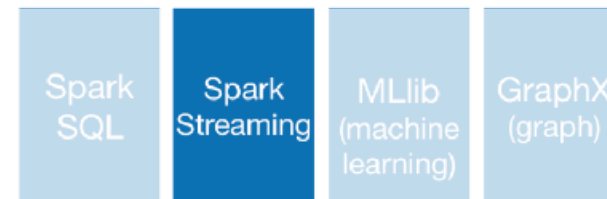
Module Spark SQL



- Principal outil pour manipuler la donnée structurée
- Accès unifié à plusieurs formats (csv, parquet, hive, ...)
- Données structurées sous format « tabulaire »:
 - Dataframe
 - Dataset
- Manipulation type SQL

Définition de Spark

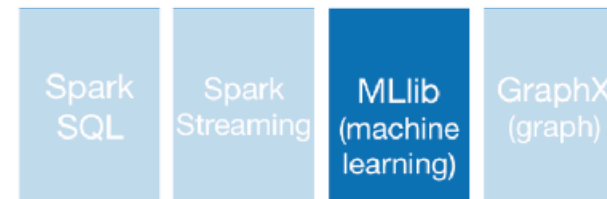
Module Spark Streaming



- Traitement de données en flux continue (micro-batch)
- Principales sources :
 - HDFS/S3
 - Kafka/Kinesis
- Applications
 - Streaming ETL / Enrichissement en temps réel
 - Détection en temps réel d'anomalie

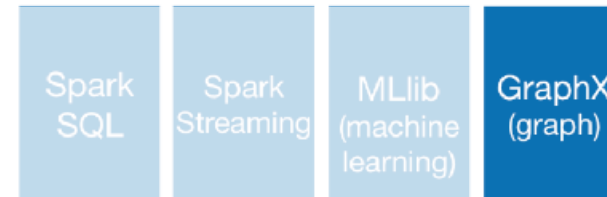
Définition de Spark

Module MLlib



- Librairie de machine learning « scalable »
 - Algorithme de ML
 - Utilitaires :
 - Mathématique et statistique
 - Feature engineering
 - ML Pipelining
 - ...

Définition de Spark Module GraphX



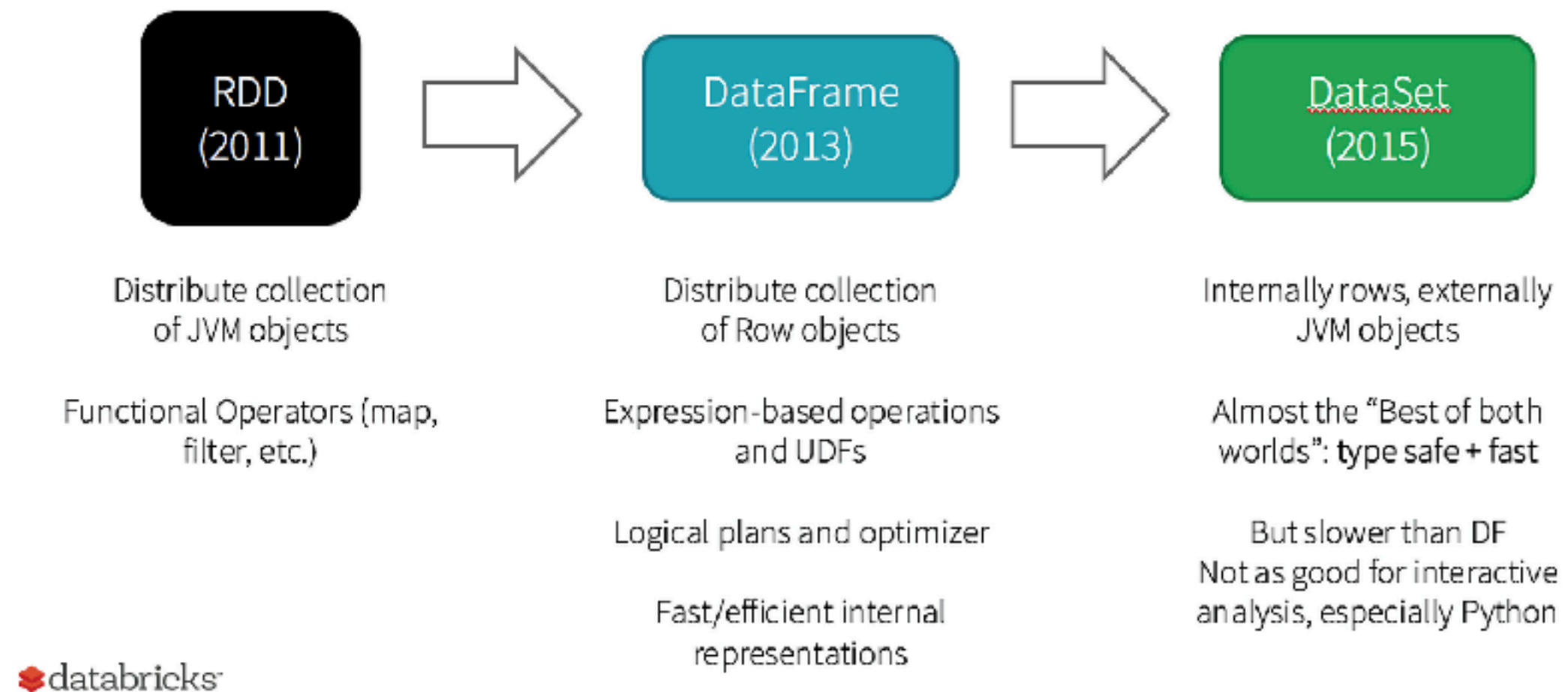
- Manipulation des graphes (basée sur RDD)

Fonctionnement de Spark

Fonctionnement de Spark

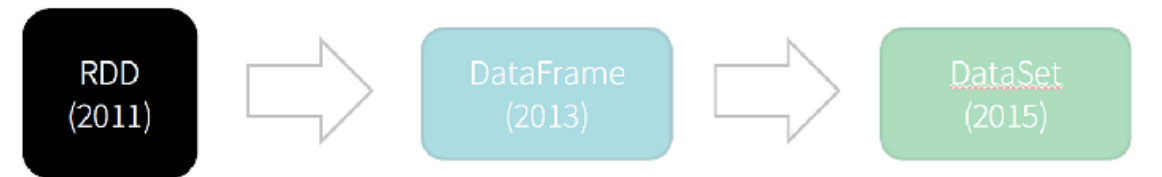
- Formats de données
 - RDD (Resilient Distributed Datasets)
 - Dataframe
 - Dataset

Fonctionnement de Spark



Fonctionnement de Spark

Format de données



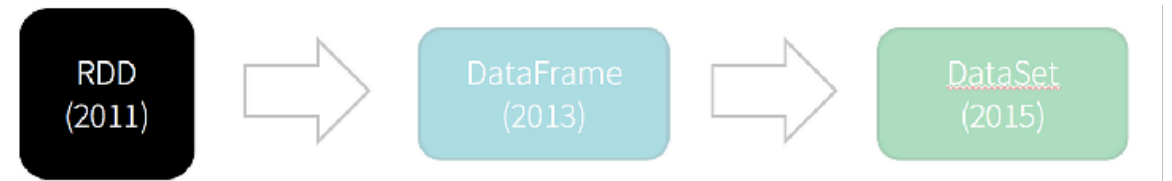
- RDD

```
>>> rdd = sc.textFile('file:/root/cours-spark/dump/large-table/1913377.csv')
>>> header = rdd.first()
>>> rdd.map(lambda x:x.split(',')).take(5)
[[u'id', u'val1', u'val2'], [u'64727', u'3', u'195'], [u'64727', u'37', u'196'],
[u'64727', u'93', u'411'], [u'64727', u'97', u'185']]
```

```
>>> rdd\
... .filter(lambda row:row != header)\
... .map(lambda row:row.split(','))\
... .map(lambda row:(int(row[0]), int(row[1])))\
... .reduceByKey(lambda a, b: a + b)\
... .collect()
[(42880, 573), (23158, 459), (75291, 581), (64727, 550), (82087, 612)]
```

Fonctionnement de Spark

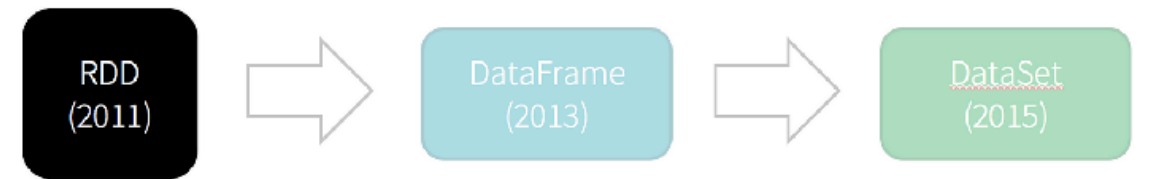
Format de données



- RDD
 - Resilient: recompute en cas d'échec
 - Distributed: partition et distribué sur plusieurs noeuds
 - Immutable: un RDD ne peut être modifié
 - Lazy: un RDD est une chaîne de calcul (transformation), mais n'exécute pas ces calculs (action)

Fonctionnement de Spark

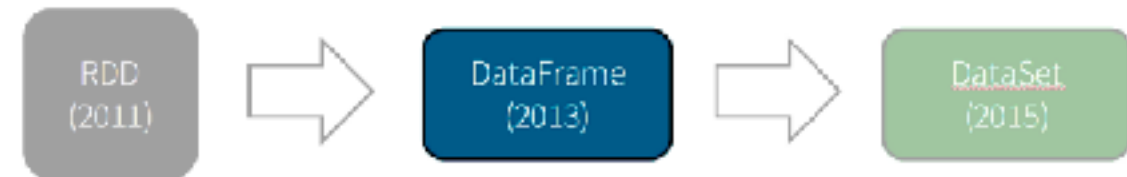
Format de données



- RDD
 - Simple
 - Fault-tolerant
 - Définition du DAG par l'utilisateur
 - (-) Pas forcément optimisé

Fonctionnement de Spark

Format de données



- Dataframe

```
df.show()
```

```
# +-----+-----+-----+
# |      id|val1|val2|
# +-----+-----+-----+
# |64727|    3| 195|
# |64727|   37| 196|
# |64727|   93| 411|
# |64727|   97| 185|
# |64727|   84| 233|
# +-----+-----+-----+
```

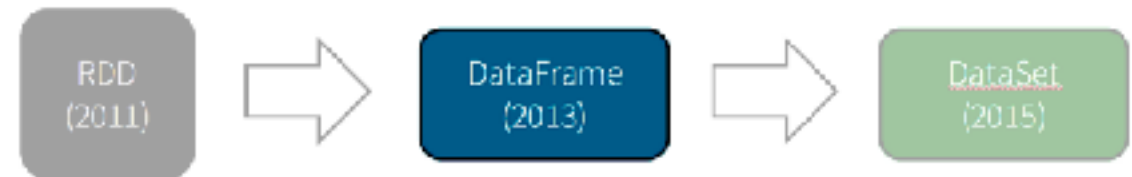
```
# only showing top 5 rows
```

```
df.rdd.take(5)
```

```
[Row(id=64727, val1=3, val2=195), Row(id=64727, val1=37, val2=196), Row(id=64727, val1=93, val2=411),  
Row(id=64727, val1=97, val2=185), Row(id=64727, val1=84, val2=233)]
```

Fonctionnement de Spark

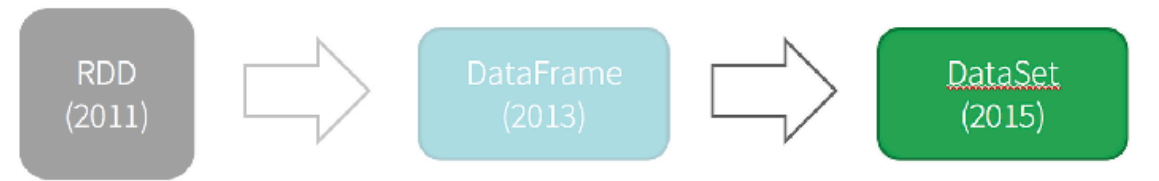
Format de données



- Dataframe
 - Format structuré tabulaire
 - Lazy
 - Amélioration de la gestion mémoire (projet Tungsten)
 - Plan d'optimisation du DAG (Catalyst)
 - (-) Impossible d'avoir les erreurs de typage avant de compiler...
(compile-time type safety)

Fonctionnement de Spark

Format de données



- Dataset
 - Disponible pour Scala/Java
(en Python —> data frame++)

Manipulation de données avec Spark

Quelques utilitaires de Spark

Interfaces avec Spark

- Terminal
- Zeppelin : <http://sandbox-hdp.hortonworks.com:9995>
- Spark submit

Suivi de performance de Spark

- Spark UI : http://sandbox-hdp.hortonworks.com:8088/proxy/application_1548959827050_0002/