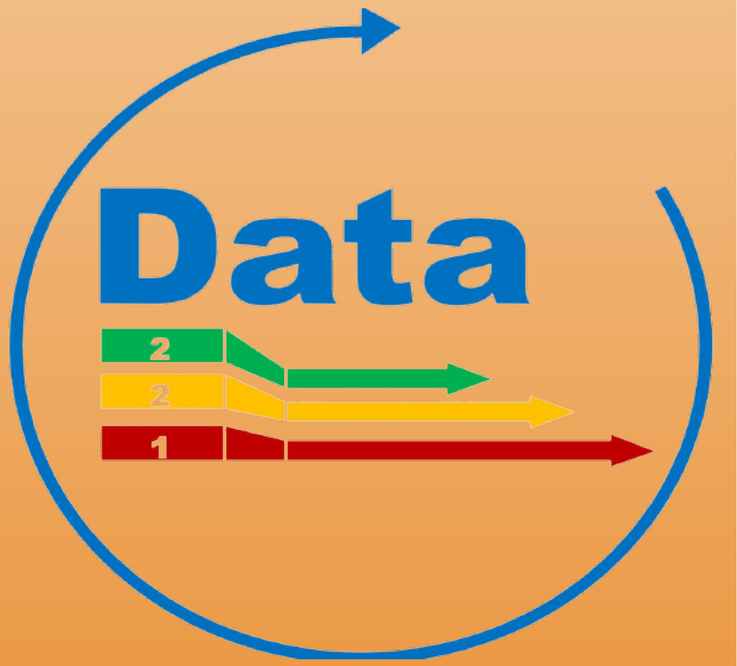


Introduction à Apache Spark

A propos de Moi

2020-Present: Big Data Engineer @ Atos Sénégal

- Aider les métiers sur le développement des KPI's.
- Mettre à disposition des data scientists, des données aggrégées pour leur modelling.

2019-2020: Big Data Engineer @ MNS Consulting

- Installation d'un environnnement Big Data et Mis en place d'une plateforme d'analyse et de prise de decision, se basant sur les données mobile.
- Mis en place de plateforme de visualization des données provenant des capteurs IOT's.

2017-2019: Master Degree en Big Data @ AIMS Senegal & Master Degree en Cryptographie @ Université de Thiès

- Msc en Big Data à AIMS (Major Promo)
- Msc Cryptographie (Mention TBien): Générateurs de nombres pseudo-aléatoires sur les résedaux euclidiens.

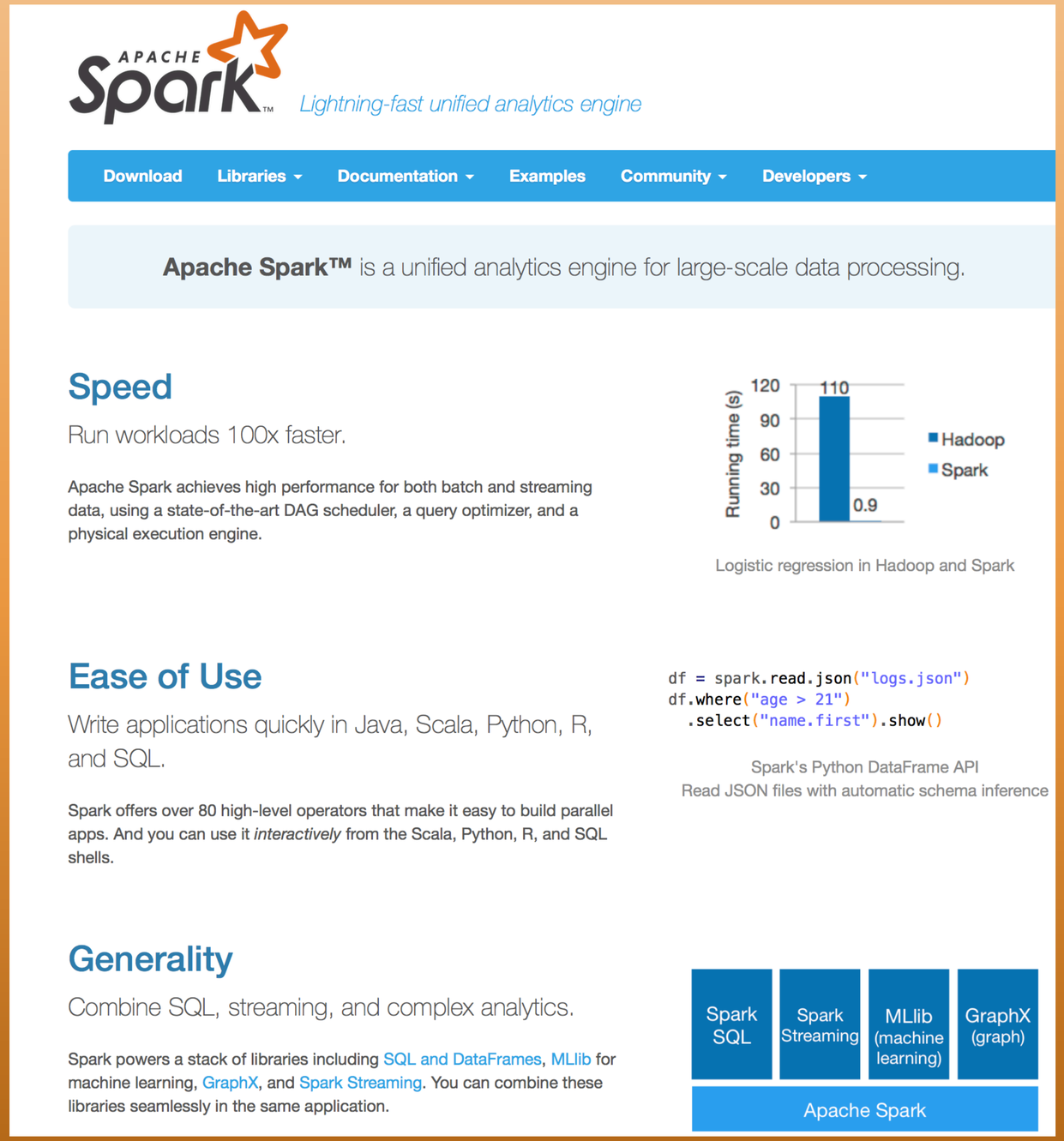
Ibrahima FALL
Consultant Big Data Engineer @ Atos Senegal,
Email: ibrahima.fall@atos.net
Alt-email: iboudofall@gmail.com

What is Spark?

Apache Spark est un moteur de traitement de données ,

- in-memory
- open-source
- ultra-rapide
- tolérant aux pannes
- supporte plusieurs languages
- Moteur unifié
- lazy computations

pour le big data et le machine learning, avec des modules pour des traitements de données en temps reel, sql et calcul de graph.



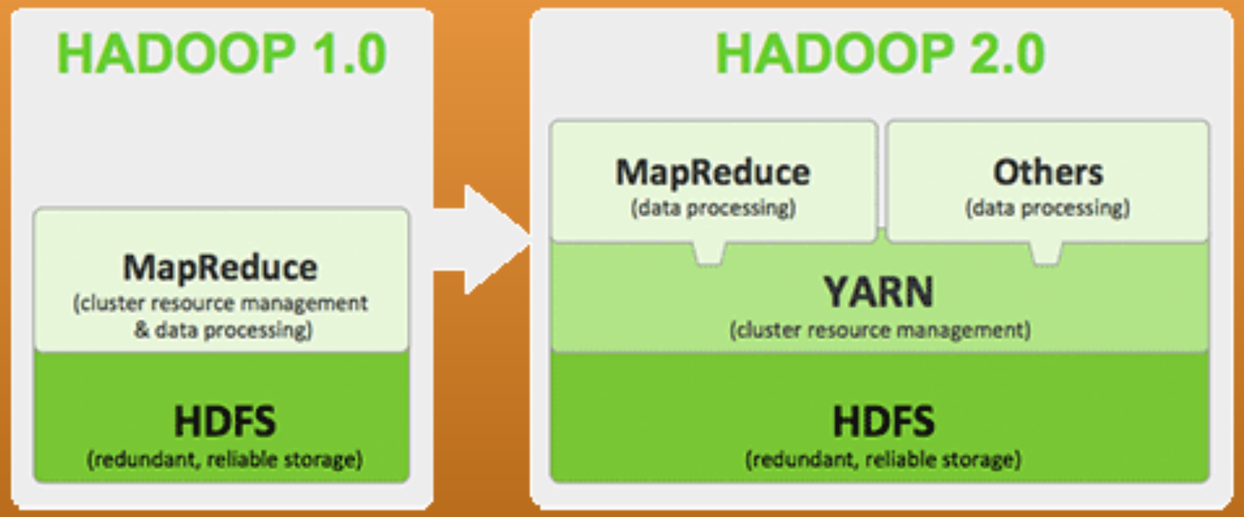
Spark vs Hadoop

Apache Hadoop

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

- Hadoop est une plateforme Big Data open-source, utilisée pour stocker et traiter d'immenses volumes de données d'une manière distribuée.
- Composants principaux:
  - HDFS (Hadoop Distributed File System)
  - YARN (Yet Another Ressource Negociator)
  - MapReduce



Spark vs Hadoop

- En réalité, Spark et comparable à MapReduce et non à Hadoop.
- Spark peut se branché à Hadoop et utiliser HDFS pour bénéficier du système de stockage de cette dernière.
- Spark est plus rapide que MapReduce (10 à 100 fois)
- Spark est écrit en Scala, MapReduce est écrit en Java
- En traitement, Spark supporte du **batch**/ **real-time**/ **iterative**/ **interactive**/ **graph** et MapReduce ne supporte que du **batch**
- MapReduce ne supporte pas du Caching, contrairement à Spark

Why Spark?

Rapidité

Facilte le développement avec les api Java, Scala, Python, R, SQL

Flexibilité

Syntaxes trés simples

Moteur unifié

Apache Spark — Lightning-fast unified analytics engine

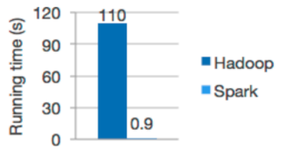
Download | Libraries | Documentation | Examples | Community | Developers

Apache Spark™ is a unified analytics engine for large-scale data processing.

Speed

Run workloads 100x faster.

Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.



Logistic regression in Hadoop and Spark

Ease of Use

Write applications quickly in Java, Scala, Python, R, and SQL.

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python, R, and SQL shells.

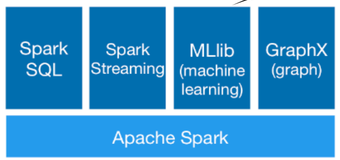
```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

Spark's Python DataFrame API
Read JSON files with automatic schema inference

Generality

Combine SQL, streaming, and complex analytics.

Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. You can combine these libraries seamlessly in the same application.



Runs Everywhere

Spark runs on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud. It can access diverse data sources.

You can run Spark using its standalone cluster mode, on EC2, on Hadoop YARN, on Mesos, or on Kubernetes. Access data in HDFS, Alluxio, Apache Cassandra, Apache HBase, Apache Hive, and hundreds of other data sources.

Utilisation de Spark

Suivant le projet, Spark peut être utilisé

- En mode single node

- En mode cluster, sur plusieurs machines

  - On Premise

  - On the cloud: Amazon Web Services(AWS), Microsoft Azure

Installation de Spark

Lien de téléchargement: https://spark.apache.org

- Pré-requis: Java8
- Télécharger une version de spark, décompresser le fichier et change quelques variables d'environnement.

Installer Spark sur un cluster en local ou sur le cloud pourrait être un peu compliqué, ainsi utiliser des plateformes tierces serait plus convénient:

- Cloudera
- Hortonworks
- MapR
- Databricks

API's de Spark

Spark APIs: RDD, Dataset et DataFrame

RDD

- RDD = Resilient Distributed Dataset (jeu de données distribué et resilient)
- RDDs sont les structures fondamentales de Spark
- RDDs sont tolerant aux pannes, immuables, partitionnés, distribués, supportent le lazy evaluation
- Compile-time Type Safe

DataFrames

- Même structure que pandas dataframe et dataframes dans R
- Représente unne abstraction des RDDs
- Supporte Catalyst Optimisation
- Erreurs syntaxiques décelées à la compilation
- Erreurs analytiques décelées au runtime

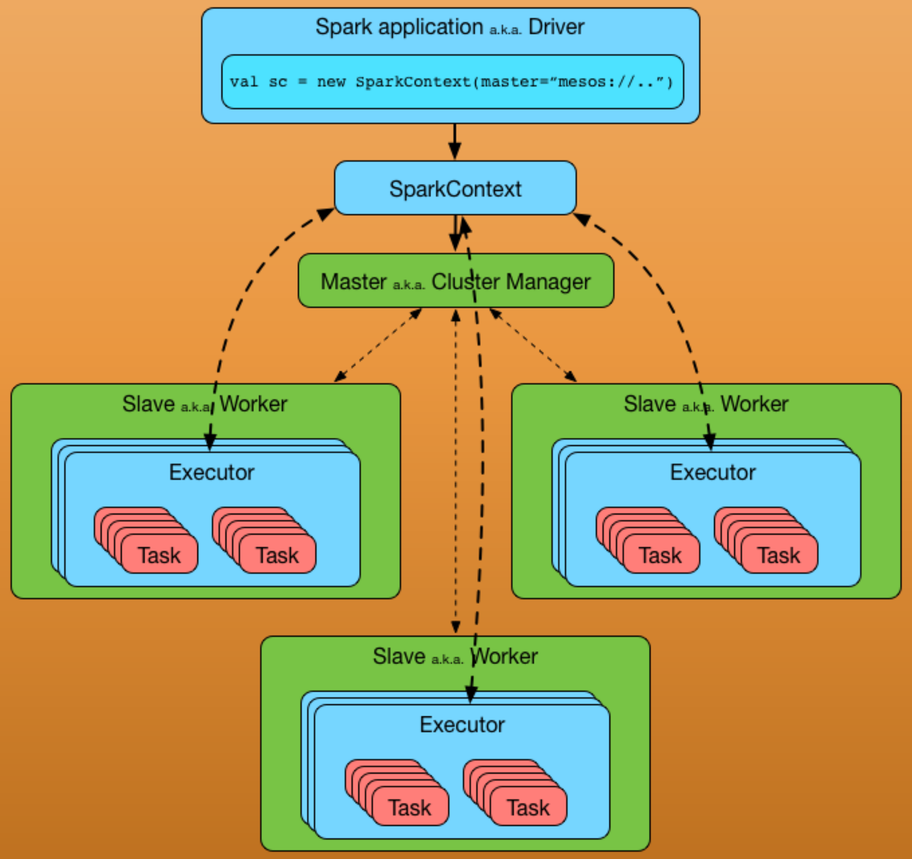
Dataset

- Un union de RDD et DataFrame

Architecture Spark

- Driver : c'est le processus contenant le programme spark, il crée le DAG, plannifie et coordonne l'execution du programme
- SparkContexte(SparkSession) : il représente la porte d'entrée du Spark Cluster avec des configurations prédéfinies (Master, nom de l'application, mémoire, nombre d'exécuteurs …)
- Master : négocie de la ressource et la rend disponible pour le driver, suit le status de ces dernières également
- Cluster Manager: suit les workers et réservent le mémoire suivant la demande du Master
- Executors : lieu d'exécutionn des tasks du DAG

Master et le Cluster Manger peuvent être fusionnés comme en mode StandAlone ou séparés comme en mode YARN



Source: https://sunbiaobiao.gitbooks.io/sa/content/spark-architecture.html

Opérations dans Spark

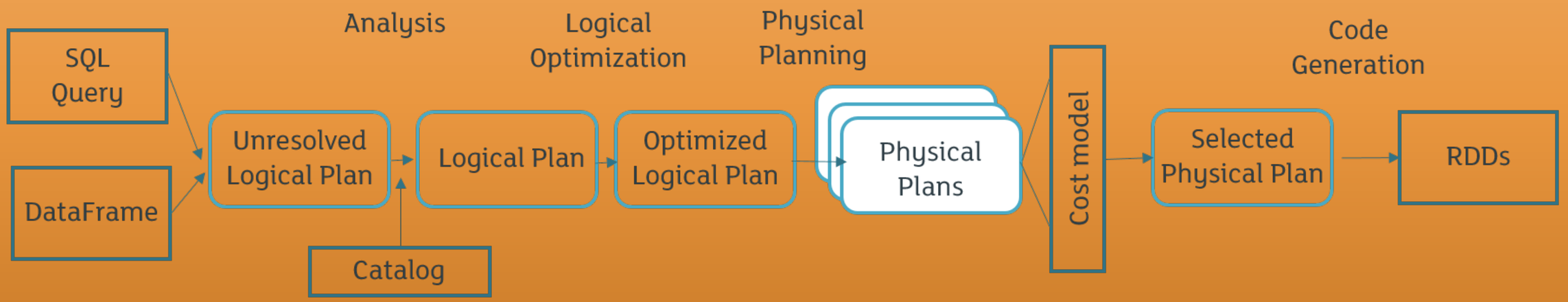
Spark dispose de deux types d’operations:

- Transformations: créer une nouvelle dataset à partir d’une autre existente
- Actions: retoune une valeur finale ou écrit le résultat dans un espace de stockage externe.

Les transformations sont dites "lazy": elles ne s’executent pas immédiatement: en effet, elles gardent juste l’ensemble des operations qui ont permit de construire la dataset résultante (lineage). Les transformations ne sont executées que lorsqu’une action est réquise.

L’ensemble des operations faites sur un dataset forme un DAG (Direct Acyclic Graph) dont les sommets sont des RDD et les arêtes représentent les operations.

Optimization



Source: https://medium.com/codex/apache-spark-optimization-techniques-and-tuning-52cc446c07f5

Fin de la présentation

Merci pour votre attention