

À la Découverte de APACHE **spark**TM

Alpha amadou DIALLO

SOMMAIRE

Le Big Data

Apache Hadoop

Apache Spark

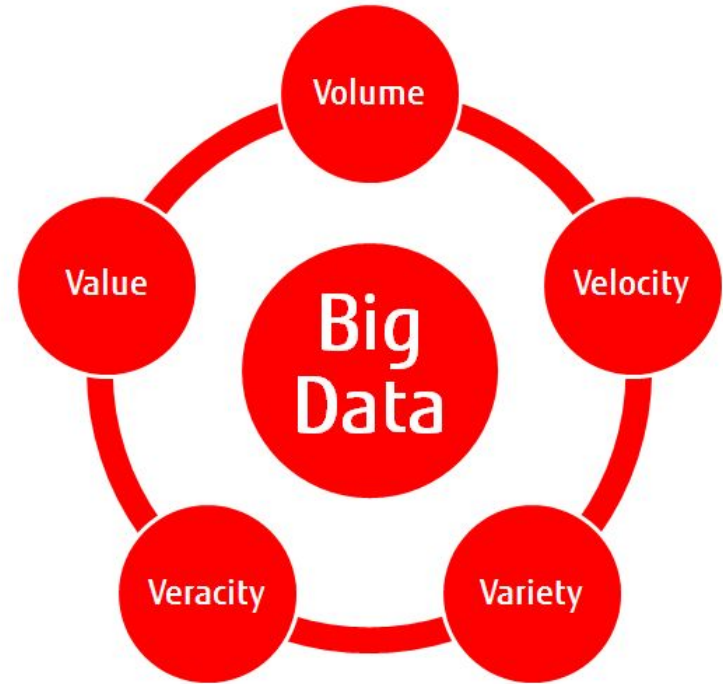
Demo

Le Big DATA

Big Data fait penser au Mégadonnées (dans l'ordre de **Po**)

Big Data = Le volume de données ?

Big Data c'est le **3V** **4V** **5V**



Le Big DATA

Explosion de la quantité des données

Le partage de données dans plusieurs serveurs

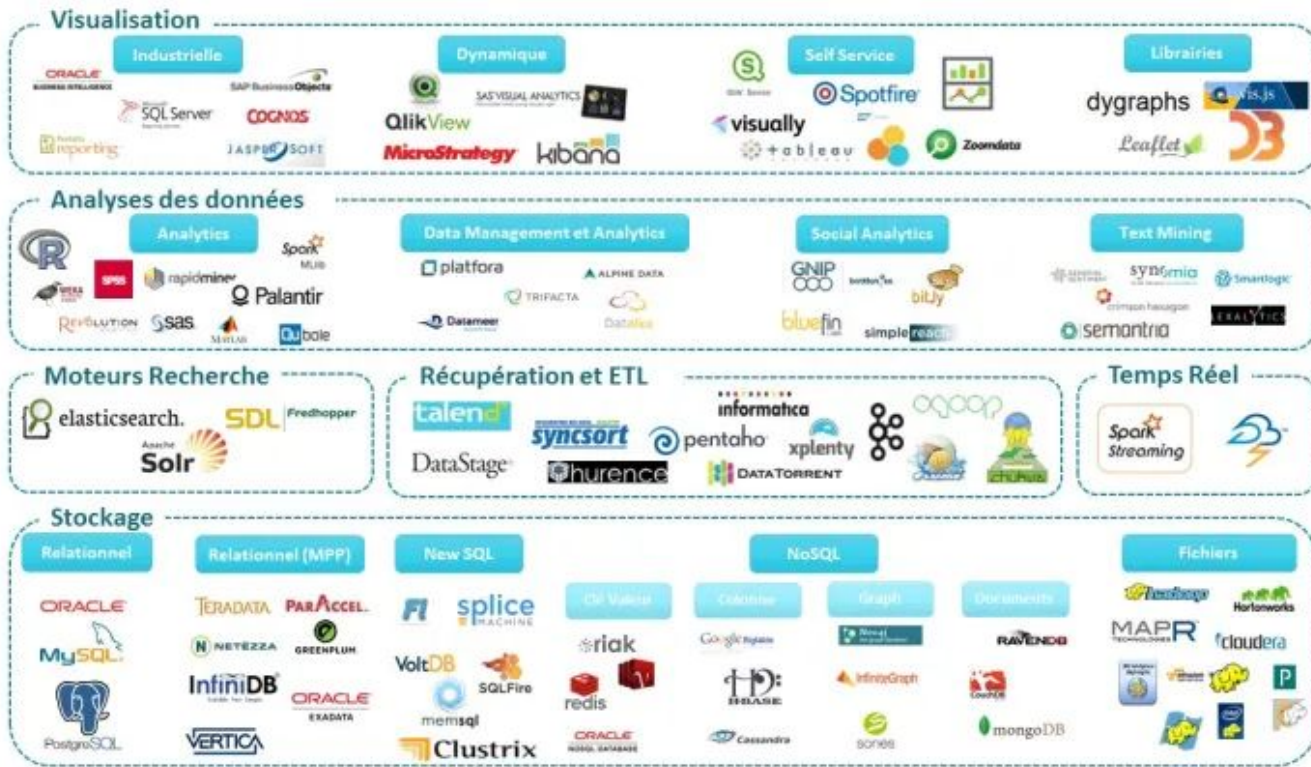
La recherche des données

Le stockage de quantités massives de données

Le traitement des flux de données



L'écosystème du Big Data



Hadoop

Conçu par **Doug Cutting**, 2004

Écrit en Java

Framework Open Source

Aide à créer des applications distribuées (dans le stockage et le traitement de données)

Résiste aux pannes

2009, il est passé à la fondation Apache



Utilisation de Hadoop

Stockage de très gros volumes de données (Po)

Les **réseaux sociaux** (Facebook, LinkedIn, Twitter)

Des sites **e-commerce** comme ebay

Analyse de fichiers **non-structurés**

etc.



HDFS

Comme **H**adoop **D**istributed **F**ile **S**ystem

Système de stockage de données

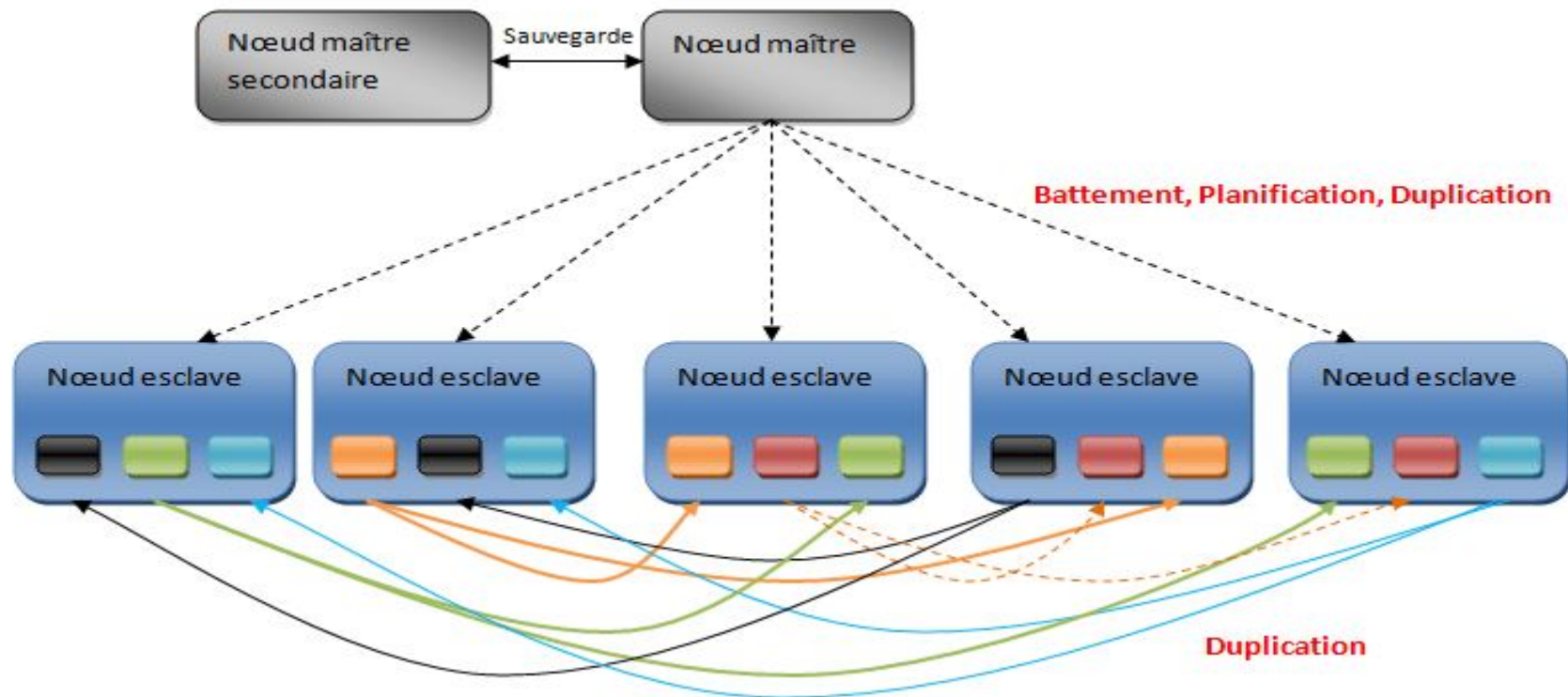
Peut stocker de grosses volumes de données

Traitement parallèle et distribué (Traitement simultané et sur plusieurs machines)

Tolérant aux erreurs (par la réplication des données)



Principe de HDFS



MapReduce

C'est un **modèle de programmation**,

Calculs parallèles et distribués sur des données très volumineuses,

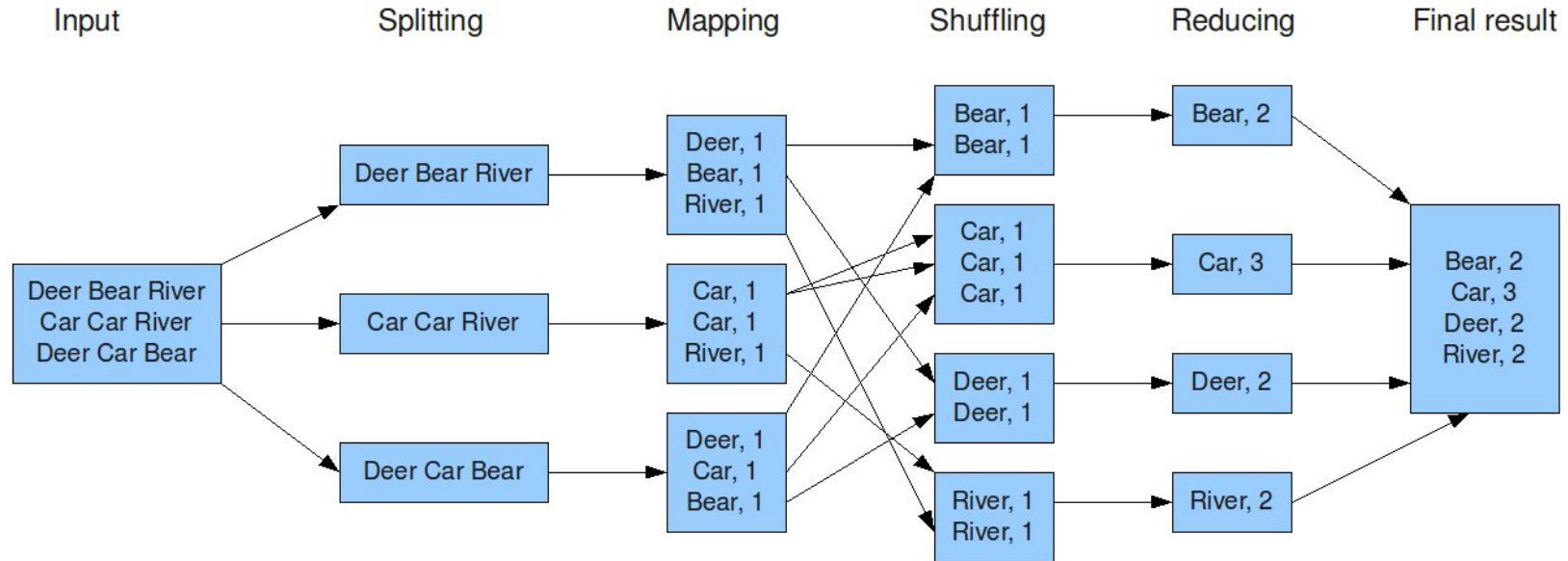
Map réalise des opérations dédiées à chaque élément
(**JavaScript** , **Python** ...),

Reduce rassemble tous ces éléments et délivre le résultat sous
forme condensée.



Exemple de MapReduce

The overall MapReduce word count process



Spark

Conçu par **Matei Zaharia**, en 2009

A l'origine, Spark est une solution pour accélérer le traitement des systèmes Hadoop

Il est écrit en **Scala**, Fonctionne sous **JVM**

Open Source sous licence BSD en 2010

2013, il est passé à la fondation Apache

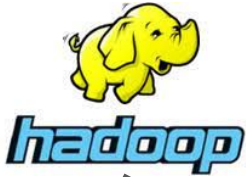
2014, vainqueur du **Daytona GraySort Contest**

Les contributeurs de son développement sont nombreux et issus des entreprises comme **Intel, Facebook, IBM, Netflix, Yahoo, Databricks** etc.

Aujourd'hui, spark a 1802 contributeurs dans github

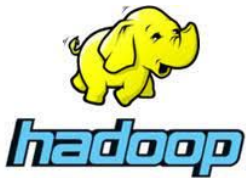


Daytona GraySort Contest



100 To

Daytona GraySort Contest



100 To



2100 Machines

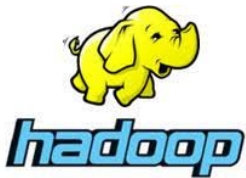


100 To



206 Machines

Daytona GraySort Contest



100 To



2100 Machines

72 minutes



100 To



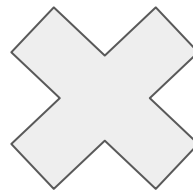
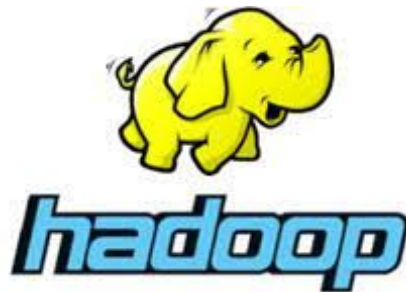
206 Machines

23 minutes

Spark vs Hadoop

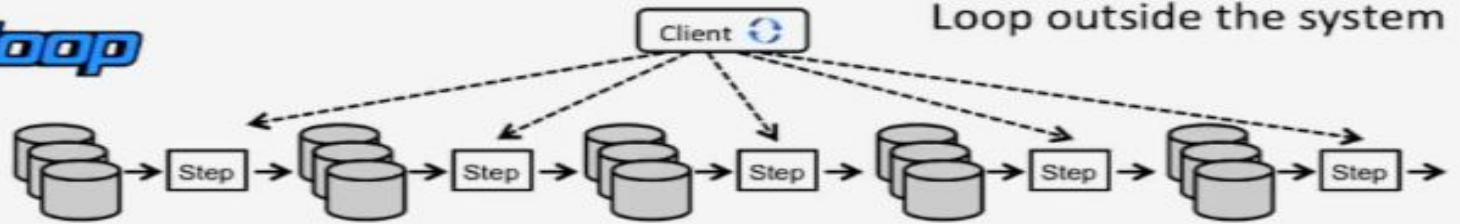


=
Vitesse

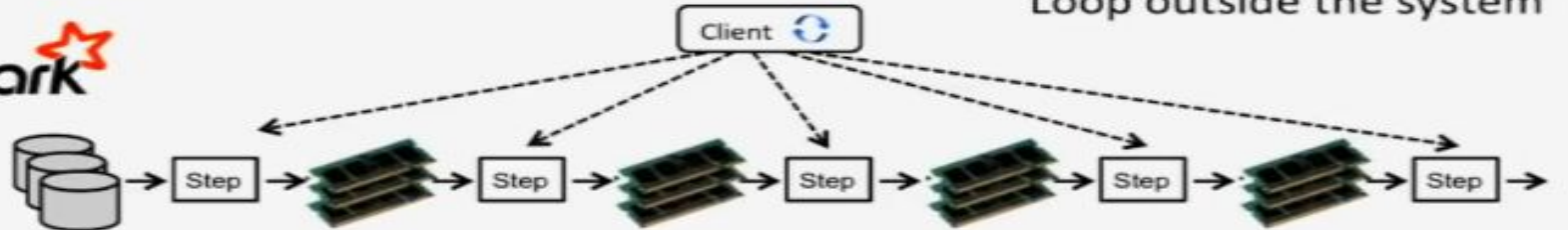


3

Spark vs Hadoop

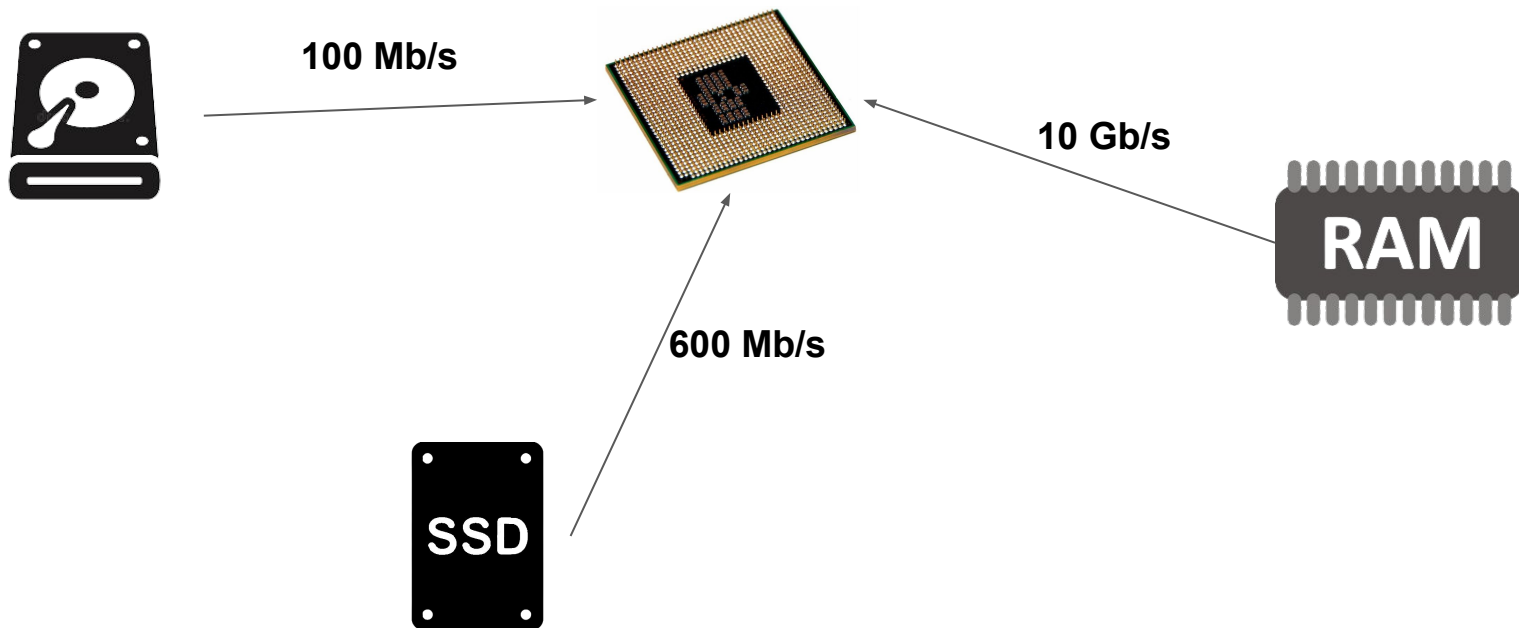


→ Move data through disk and network (HDFS)



→ User can cache data in memory

Taux de transfert de données



Spark

C'est un **framework** qui est utilisé pour les systèmes Big Data

Assure un traitement **parallèle** et **distribué** des données massives

Permet un ensemble de traitement (**Streaming, Batch**)

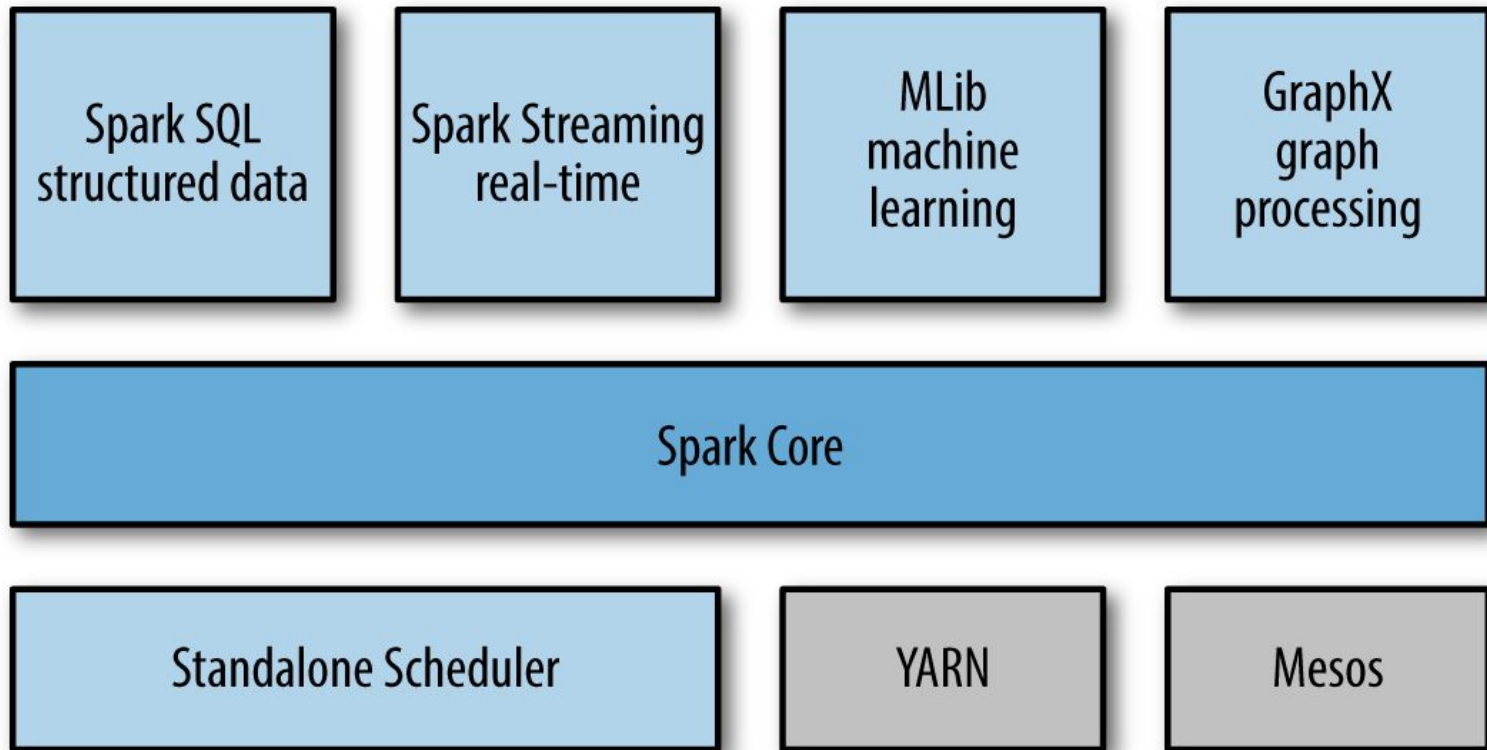
Spark est une technologie de **traitement**

Spark peut s'exécuter dans Hadoop **HDFS**

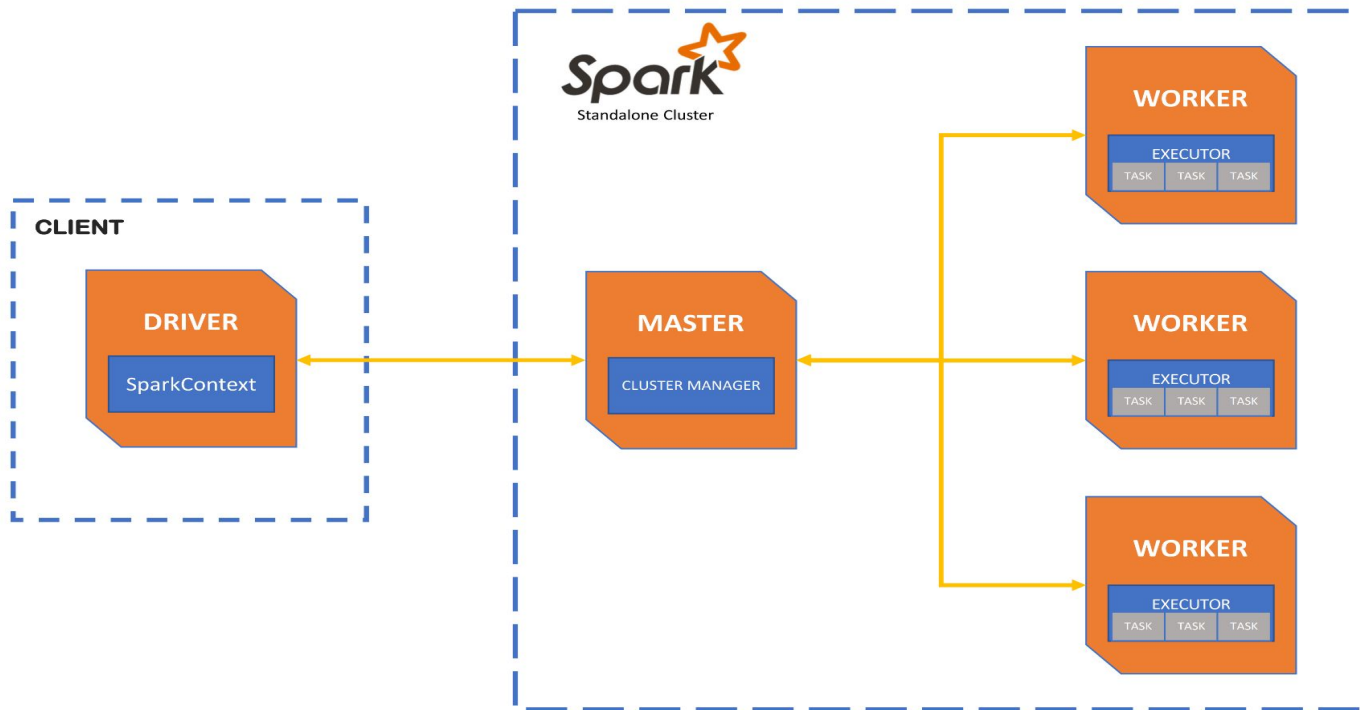
Offre des APIs de haut niveau en Java, Scala, Python et R.



Composants de Spark



Architecture de Spark



Caractéristiques de Spark

Performance de traitement

Tolérance aux pannes (grâce aux **RDD**)

Traitement à la **volée** (comparé à MapReduce qui ne traite en Batch)

Support de plusieurs langages

Une **communauté** active et en expansion

Intégration avec Hadoop (grâce aux gestion de ressources avec YARN)

Fonctionne en **mémoire**



limites de Spark

Problèmes avec les fichiers de petite taille

Traitement en temps réel

Pas de système de stockage

Peu d'Algorithmes ML

Coûteux (Traitement en mémoire)



Spark et le système de stockage



Applications utilisant Spark

Les ETL

Analyse prédictive et Machine Learning

Opérations d'accès aux données (SQL)

Traitement et extraction de texte

Traitement temps réel

Applications graphiques



Qui utilise Spark ?



Data Analyste



Data Scientists



Data Engineer



Ingénieurs MLOps

Resilient Distributed Datasets (RDD)

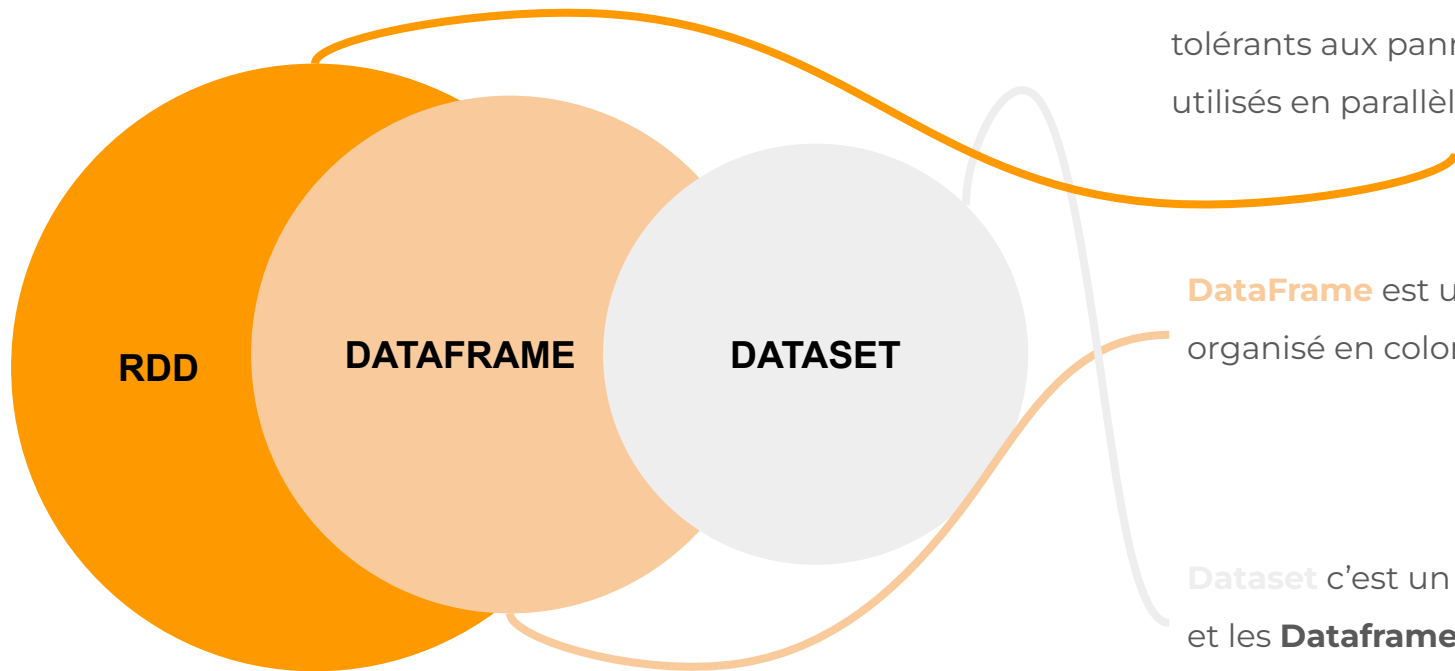
Un RDD est une collection de données calculée à partir d'une source et conservée en mémoire vive.

Les RDD doivent suivre quelques propriétés telles que:

- Immuable,
- Tolérance de panne,
- Distribué

```
1 Ndambé,Fall,34
2 Lamarana,DIALLO,12
3 Petit Yero,DIIOUF,34
4 Sambay,Bathie,45
```

APIs de Spark



RDD est une collection d'éléments tolérants aux pannes pouvant être utilisés en parallèle.

DataFrame est un jeu de données organisé en colonnes nommées

Dataset c'est un peu l'Union des **RDDs** et les **Dataframes**

Télécharger & Installer de Spark

Ce rendre sur ce lien : <https://spark.apache.org/downloads.html>

Download Libraries ▾ Documentation ▾ Examples Community ▾ Developers ▾

Download Apache Spark™

1. Choose a Spark release:
2. Choose a package type:
3. Download Spark: [spark-3.3.0-bin-hadoop3.tgz](#)
4. Verify this release using the 3.3.0 [signatures](#), [checksums](#) and [project release KEYS](#) by following these [procedures](#).

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Vous pouvez également regarder cette vidéo pour plus de détails: <https://www.youtube.com/watch?v=JQ7HOilu2iq&t=270s>

Télécharger & Installer Spark



```
sudo apt update
```

```
//Installer JAVA
```

```
sudo apt install default-jre
```

```
sudo apt install default-jdk
```

```
sudo apt update
```

```
//Installer curl (Optionnel)
```

```
sudo apt install curl
```

```
//Télécharger spark version 3
```

```
curl -O https://d1cdn.apache.org/spark/spark-3.3.0/spark-3.3.0-bin-hadoop3.tgz
```

```
tar xvf spark-3.3.0-bin-hadoop3.tgz
```

```
//Déplacer le contenu dans spark
```

```
sudo mkdir /opt/spark
```

```
sudo mv spark-3.3.0-bin-hadoop3/* /opt/spark/
```

Ajouter les variables d'environnement



```
source .bashrc
```

1

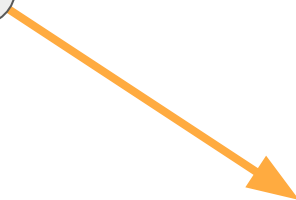
```
#Ajouter les lignes ci-dessous dans le fichier bashrc  
export $SPARK_HOME = /opt/spark  
export PATH = $PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin  
#Save avant de quitter le fichier bashrc
```

2



```
nano .bashrc
```

3



Tester Spark

```
alpha@alpha-VirtualBox:~$ spark
spark-class          spark-daemon.sh    spark-shell
spark-class2.cmd     spark-daemons.sh   spark-sql
spark-config.sh      sparkR              spark-submit
alpha@alpha-VirtualBox:~$ spark
```




Spark dans le cloud (Databricks)

<https://databricks.com/try-databricks> (Pour créer un compte)

<https://databricks.com/> (Site Officiel)

Try Databricks for free







An open and unified data analytics platform for data engineering, data science, machine learning, and analytics. From the original creators of Apache Spark™, Delta lake, MLflow, and Koalas.



Databricks trial:

- Collaborative environment for data teams to build solutions together.
- Interactive notebooks to use Apache Spark™, SQL, Python, Scala, Delta Lake, MLflow, TensorFlow, Keras, Scikit-learn and more.
- Available as a 14-day full trial in your own cloud, or as a lightweight trial hosted by Databricks.

Used by:



Please tell us about yourself

First Name: *

Last Name: *

Company *

Company Email *

Title *

Phone Number

Country: *

Senegal ▼

By submitting, I agree to the processing of my personal data by Databricks in accordance with our [Privacy Policy](#). I understand I can update my preferences at any time.

GET STARTED FOR FREE

Wordcount avec Scala



```
scala> val lines = sc.textFile("./Discours.txt")

scala> val words = lines.flatMap(_.split("\\s+"))

scala> val wc = words.map(w => (w,1)).reduceByKey(_ + _)

scala> wc.saveAsTextFile("./word_count")
```

Wordcount avec Python



```
#pip install pyspark
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("WordCount").getOrCreate()

words = spark.sparkContext.textFile("./Discours.txt").flatMap(lambda line: line.split(" "))

wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)

wordCounts.saveAsTextFile("Words Count")

print("Fin du programme !")
```

Ressources

Cours de Alphorm

https://www.youtube.com/watch?v=_blZ3g2kb9c&list=PL1aYsXmhJ1Wf1B6Zm8SpK863jH0jiqytB

Cours de TechWall

https://www.youtube.com/watch?v=inoRa9TbX_M&list=PLI3CtU4THqPaHpU5g1-iFyT72SACUmCv

Plateforme Databricks

<https://databricks.com/>

Documentation Officiel de Spark

<https://docs.databricks.com/getting-started/introduction/index.html>