Apache Spark: Une Exploration

Présentation d'Apache Spark, un moteur de traitement de données rapide et puissant. Il est utilisé pour le Big Data.

Qu'est-ce qu'Apache Spark ?

Spark est un framework de traitement distribué. Il traite des données en mémoire.

Il offre une rapidité et une flexibilité exceptionnelles.

Rapidité

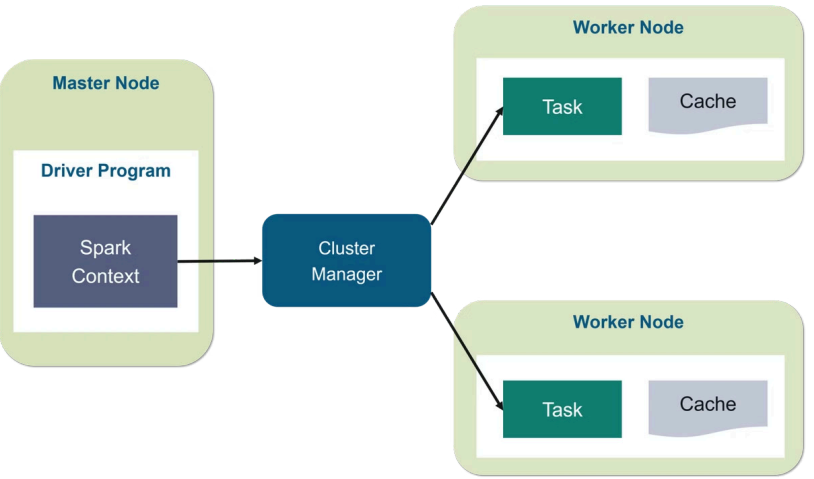
Traitement en mémoire, optimisé.

Polyvalence

SQL, streaming, machine learning.

Scalabilité

S'adapte à des volumes massifs.



Spark et les 5V du Big Data

Spark excelle dans le traitement des données volumineuses (Volume).

Il gère les données variées (Variety) et la vélocité (Velocity).

Il assure la fiabilité (Veracity) et la valeur (Value).

Volume

Gère des pétaoctets de données.

Vélocité

Traitement en temps réel ou quasi-réel.

Variety

Données structurées, semi-structurées et non structurées.

Historique d'Apache Spark

Créé à l'Université de Californie, Berkeley.

Projet open source, communauté active.

Évolutions constantes, nouvelles fonctionnalités.

1

2009

Début du projet.

2

2010

Premières versions.

3

2014

Intégration dans Hadoop.

4

2020

Versions récentes.

Liens avec les Entreprises

Utilisé par de nombreuses entreprises.

Amazon, Google, Microsoft, Databricks.

Intégration dans les plateformes cloud.

1 **Amazon**

Intégré à AWS.

2 **Google**

Utilisé sur Google Cloud.

3 **Microsoft**

Intégré à Azure.

4 **Databricks**

Plateforme Spark.

Applications de Spark

Traitement de données en temps réel.

Analyse prédictive, machine learning.

Traitement de flux de données.

Analyse

Exploration de données.

ML

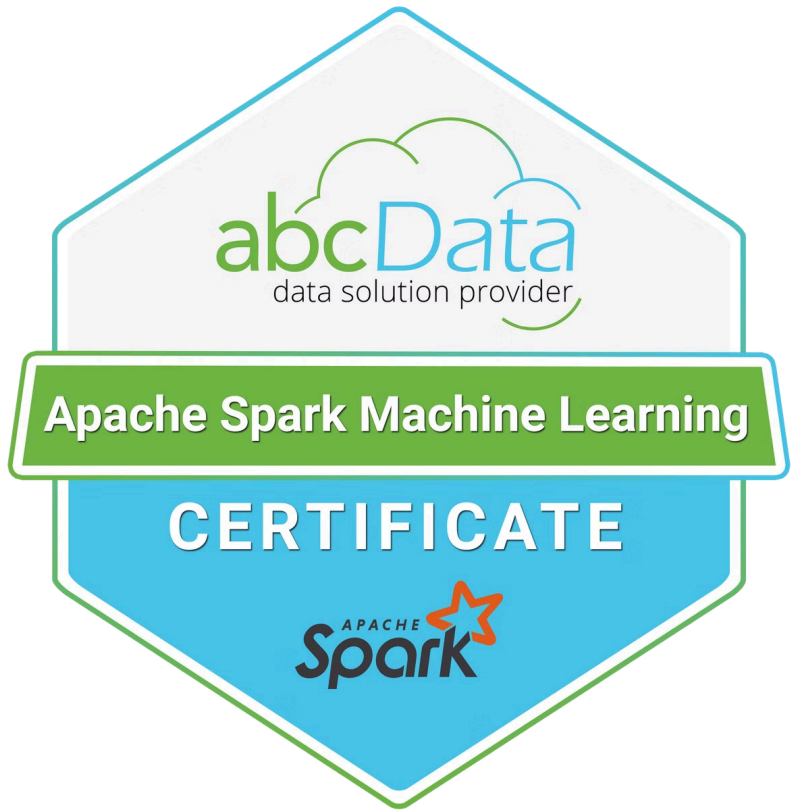
Modèles prédictifs.

Streaming

Traitement en temps réel.

SQL

Requêtes sur données.



Example d'Application Apache Spark : Installation

```
C:\Users\goulv\Documents\spark-3.5.3-bin-hadoop3>pyspark
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
24/10/13 22:52:05 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: H
ADOOP_HOME and hadoop.home.dir are unset. -see https://wiki.apache.org/hadoop/WindowsProblems
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/10/13 22:52:06 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.3
      /_/

Using Python version 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021 19:00:18)
Spark context Web UI available at http://Goulven.mshome.net:4040
Spark context available as 'sc' (master = local[*], app id = local-1728852727630).
SparkSession available as 'spark'.
>>> 24/10/13 22:52:21 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC),
 users should configure it(them) to spark.eventLog.gcMetrics.youngGenerationGarbageCollectors or spark.eventLog.gcMetric
s.oldGenerationGarbageCollectors
```

spark.apache.org

Downloads | Apache Spark

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.

Example d'Application Apache Spark : Shuffle Benchmark

Le Shuffle Benchmark met l'accent sur les opérations PySpark courantes sur les trames de données qui déclencheraient un brassage. Les opérations de tests comprennent :

- Group By and Aggregate
- Repartition
- Inner Join
- Broadcast Inner Join
- Each operation is timed independently.

```
24/10/11 22:49:55 INFO __main__:
24/10/11 22:49:55 INFO __main__: ******************************************************************
24/10/11 22:49:55 INFO __main__:     RESULTS    RESULTS    RESULTS    RESULTS    RESULTS    RESULTS
24/10/11 22:49:55 INFO __main__:     Test Run = run-shuffle-benchmark
24/10/11 22:49:55 INFO __main__:
24/10/11 22:49:55 INFO __main__: Group By test time          = 123.87005627900362 seconds
24/10/11 22:49:55 INFO __main__: Repartition test time       = 247.30380326602608 seconds (200 partitions)
24/10/11 22:49:55 INFO __main__: Inner join test time        = 373.45172010397073 seconds
24/10/11 22:49:55 INFO __main__: Broadcast inner join time   = 308.29632237099577 seconds
24/10/11 22:49:55 INFO __main__:
24/10/11 22:49:55 INFO __main__: ******************************************************************
```

Example d'Application Apache Spark : CPU Benchmark

Le CPU Benchmark met l'accent sur les opérations PySpark qui doivent être principalement liées au CPU. Cela ne veut pas dire qu'il y aurait des E/S disque ou réseau, mais simplement que la vitesse du processeur et l'efficacité des tâches devraient être les principaux facteurs de performance du benchmark. Les opérations de test pour ce benchmark comprennent :

- Hachage SHA-512 d'une chaîne
- Estimation de Pi avec des échantillons aléatoires et une fonction Python définie par l'utilisateur
- Estimation de Pi avec des échantillons aléatoires et des fonctions Spark natives uniquement

Chaque opération est chronométrée indépendamment.

```
24/10/11 04:33:58 INFO __main__: ********************************************************************************
24/10/11 04:33:58 INFO __main__:     RESULTS    RESULTS    RESULTS    RESULTS    RESULTS    RESULTS
24/10/11 04:33:58 INFO __main__:     Test Run = pyspark-benchmark-cpu
24/10/11 04:33:58 INFO __main__:
24/10/11 04:33:58 INFO __main__: SHA-512 benchmark time                = 1021.0458517669999 seconds for 2,000,000,000 hashes
24/10/11 04:33:58 INFO __main__: Calculate Pi benchmark                = 636.8594892990002 seconds with pi = 3.14151414112, samples = 25,000,000,000
24/10/11 04:33:58 INFO __main__: Calculate Pi benchmark using dataframe = 9.309087140999964 seconds with pi = 3.14159186944, samples = 25,000,000,000
24/10/11 04:33:58 INFO __main__:
24/10/11 04:33:58 INFO __main__: ********************************************************************************
```

***Une erreur d'E/S est donc un problème avec le disque dur externe qui empêche Windows de lire son contenu ou d'écrire sur celui-ci.**

Conclusion

Spark est un outil puissant et polyvalent.

Il révolutionne le traitement du Big Data.

De nombreuses applications possibles.

Avantages	Inconvénients
Rapidité	Complexité
Scalabilité	Ressources
Polyvalence	Expertise