HYONTA KENGAP BLERIOT

Rapport De Projet

2025

Introduction à Apach Spark

Construction d'un lac de données avec Apache Spark

MASTER II
DSAD

Enseignant: M. MOUNGOLE ARISTIDE

SOMMAIRE

Résumé

Un lac de données est une architecture de stockage permettant de centraliser de grandes quantités de données brutes issues de sources variées [1]. Dans le contexte du Cameroun, l'essor des données numériques, notamment dans les secteurs de la finance, de la santé et du commerce, nécessite une solution efficace pour l'ingestion, le stockage et l'analyse des données massives. Ce projet vise à construire un lac de données en utilisant **Apache Spark**, une plateforme de traitement distribuée, tout en exploitant des bases de données relationnelles et des fichiers hétérogènes.

Les analyses effectuées sur ces données seront visualisées sous forme de graphiques et tableaux de bord interactifs, afin de faciliter l'interprétation des tendances et des anomalies au sein des différentes sources de données.

I / INTRODUCTION

I.1 Contexte et Problématique

La croissance exponentielle des données pose un défi majeur aux entreprises et institutions camerounaises [2]. Les systèmes traditionnels de gestion de bases de données (à l'exemple de MySQL et PostgreSQL) montrent leurs limites face à la variété et au volume croissant des données collectées. Dans des secteurs comme la finance, l'administration publique et le commerce, il devient crucial de centraliser ces informations pour une prise de décision rapide et efficace.

Ainsi, l'adoption des **lacs de données** (Data Lakes) devient une solution stratégique pour stocker et traiter efficacement des données structurées et non structurées [3]. Ce projet vise à construire un lac de données permettant l'intégration, le nettoyage, l'analyse et la visualisation des informations issues de différentes sources.

I.2 Objectifs du Projet

Ce projet a pour objectifs :

- De concevoir un **lac de données** capable de stocker des informations brutes hétérogènes issues de bases SQL, de fichiers logs et d'avis clients.
- De mettre en place un **pipeline d'ingestion et de transformation** avec Apache Spark.
- D'analyser ces données en vue de produire des **tableaux de bord interactifs** pour faciliter l'aide à la décision.

II / QU'EST-CE QU'UN LAC DE DONNÉES ?

Un lac de données est un système de stockage de données brutes, non transformées, provenant de sources variées (bases de données relationnelles, logs, fichiers CSV, JSON, etc.) [4]. Il présente plusieurs caractéristiques principales :

- **Stockage massivement distribué** : Prise en charge de gros volumes de données.
- **Données de diverses natures** : Structurées, semi-structurées et non structurées.
- **Flexibilité analytique** : Les données peuvent être analysées sans schéma prédéfini [5].

III / TECHNOLOGIES UTILISÉES

Le projet repose sur les technologies suivantes :

- **Apache Spark** : Pour le traitement des données distribuées.
- **MySQL & PostgreSQL** : Stockage relationnel des données transactionnelles.
- **Parquet & JSON** : Formats de stockage performants pour le lac de données.
- **Matplotlib & Pandas** : Analyse et visualisation des données en Python.
- **Jupyter Notebook** : Environnement interactif pour l'analyse et la présentation des résultats.

IV / ARCHITECTURE ET MISE EN ŒUVRE

IV.1 Ingestion des Données

Nous avons collecté des données provenant de différentes sources :

- **Données SQL** : Extraction des clients et transactions depuis MySQL/PostgreSQL.
- **Logs applicatifs** : Données textuelles issues des journaux d'activité.
- **Avis clients** : Données textuelles provenant de sources externes (ex. feedback e-commerce).

chargement des données SQL avec Spark :

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("DataLake").getOrCreate()

clients_df = spark.read.format("jdbc").options(
        url="jdbc:mysql://localhost:3306/ecommerce",
        driver="com.mysql.jdbc.Driver",
        dbtable="clients",
        user="root",
        password="password"
).load()
```

IV.2 Nettoyage et Transformation

Les données brutes ont été nettoyées en supprimant les valeurs nulles et en normalisant certains champs.

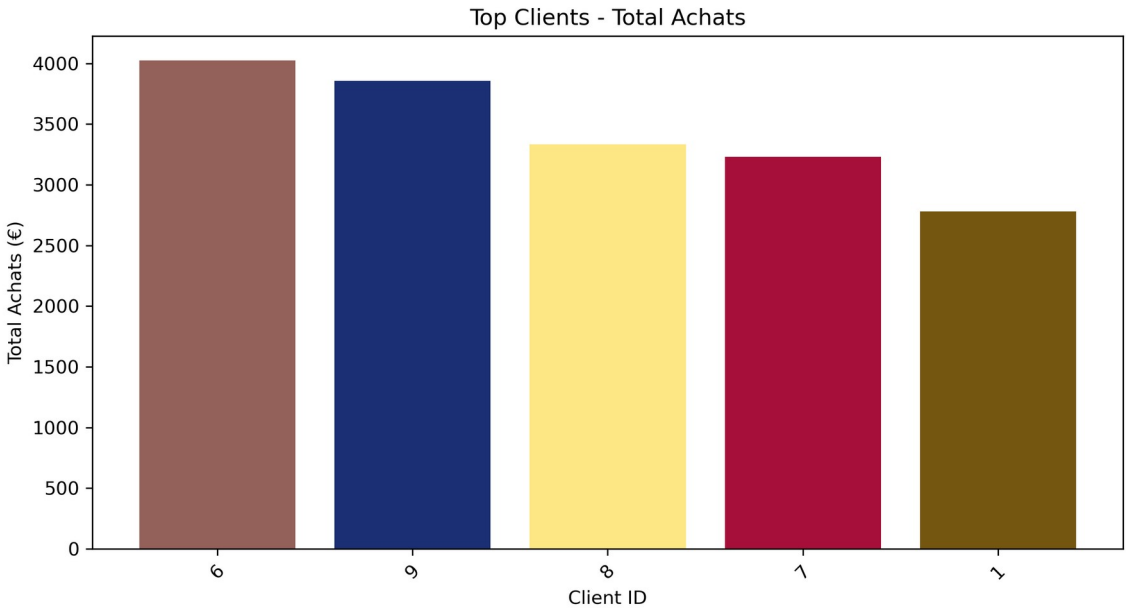
```
from pyspark.sql.functions import col, to_date

clients_clean_df = clients_df.dropna().withColumn("email", col("email").lower())
transactions_clean_df = transactions_df.dropna().withColumn("date_achat", to_date(col("date_achat"), "yyyy-MM-dd"))
```

IV.3 Analyse et Visualisation

Nous avons produit plusieurs visualisations à partir des données nettoyées

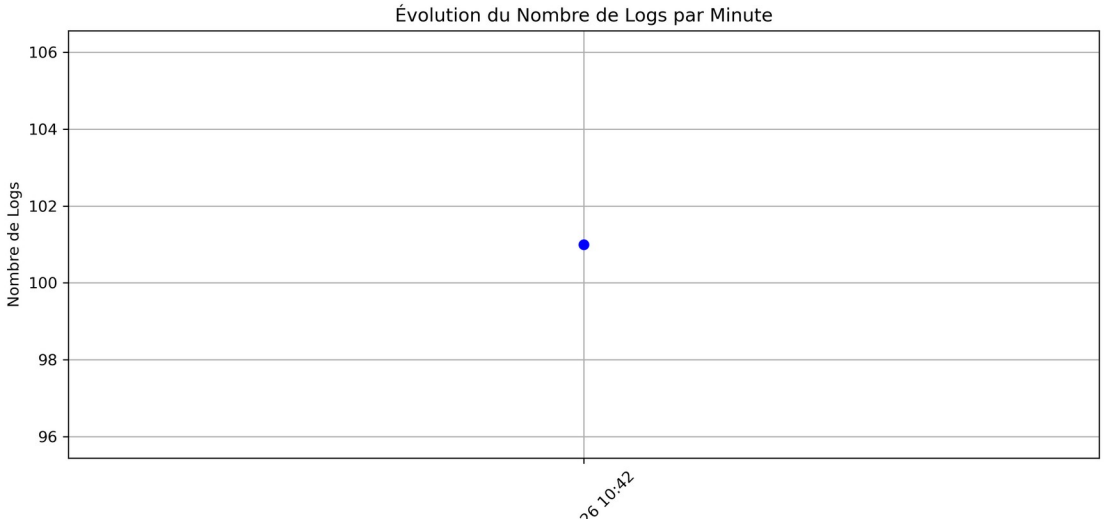
IV.3.1 Top 5 des clients ayant effectué le plus d'achats

```
plt.figure(figsize=(10, 5))
plt.bar(top_clients_pd["client_id"].astype(str), top_clients_pd["total_achat"], color=['blue', 'orange', 'green', 'red', 'purple'])
plt.title("Top Clients - Total Achats")
plt.xlabel("Client ID")
plt.ylabel("Total Achats (€)")
plt.savefig("top_clients.png")
plt.show()
```



IV.3.2 Évolution des logs d'activité par minute

```
plt.figure(figsize=(12, 5))
plt.plot(logs_par_minute_pd["minute"], logs_par_minute_pd["count"], marker='o', linestyle='-',
color="blue")
plt.title("Évolution du Nombre de Logs par Minute")
plt.xlabel("Minute")
plt.ylabel("Nombre de Logs")
plt.grid()
plt.savefig("logs_par_minute.png")
plt.show()
```



V / CONCLUSION ET PERSPECTIVES

Ce projet a permis de démontrer l'efficacité d'un lac de données dans le traitement de données volumineuses et hétérogènes en exploitant Apache Spark. L'approche adoptée a permis d'ingérer, nettoyer et analyser efficacement des données issues de sources variées telles que des bases SQL, des fichiers logs et des avis clients. L'intégration des visualisations interactives dans un tableau de bord offre un support pertinent pour l'analyse décisionnelle.

À l'avenir, nous envisageons d'améliorer l'architecture en intégrant des technologies comme Hadoop et Kafka pour la gestion des flux de données en temps réel. De plus, l'automatisation du pipeline de traitement avec Apache Airflow permettra une gestion plus efficace des processus d'ingestion et de transformation. Enfin, la mise en place d'un moteur de recherche avancé sur les logs et une interface web intuitive renforceront l'accessibilité et la performance du lac de données. Ces perspectives garantiront un meilleur usage des données dans un cadre stratégique et opérationnel au Cameroun.

VI / RÉFÉRENCES

[1] James Dixon, "Pentaho, Big Data and the Birth of Data Lakes", 2010.
[2] Ministère des Postes et Télécommunications du Cameroun, "Stratégie numérique 2020", 2019.
[3] D. Gandomi & M. Haider, "Beyond the Hype: Big Data Concepts, Methods, and Analytics", International Journal of Information Management, 2015.
[4] M. Armbrust et al., "Delta Lake: High-Performance ACID Table Storage", VLDB, 2020.
[5] Apache Spark Documentation, "Spark SQL and DataFrames", 2023.

source code: https://github.com/hyontnick/Data_lake_with_apache_spark