

TRANSLATING PYSPARK PIPELINES TO DASK

ASSISTANT
CODE



Sommaire

- 1 INTRODUCTION**
- 2 PLAN**
- 3 CONCLUSION**

Introduction



Automatisation ETL avec IA

- Oluwaferanmi, JKA (2025) – Automating ETL Pipelines Using Artificial Intelligence:

Utilisation de l'intelligence artificielle pour transformer les pipelines ETL traditionnels en workflows intelligents et adaptatifs.

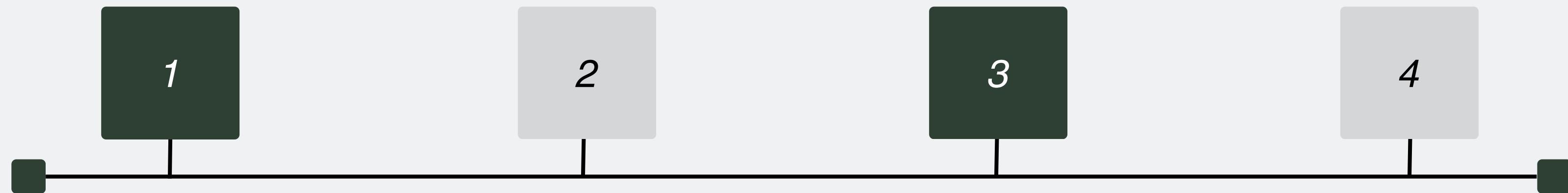
Apache Spark est largement utilisé pour le traitement de données massives, mais il reste lourd et peu intégré à l'écosystème Python.

Dask, plus léger et flexible, offre une alternative mieux adaptée aux workflows modernes de data science.

Pourtant, la migration Spark → Dask est peu documentée et complexe.

Ce projet cherche à répondre à cette problématique en proposant une méthodologie outillée (profilage, IA, benchmarks) pour faciliter et comparer cette migration

Plan



Extraire des bouts de code

PySpark

Ne pas prendre un énorme projet entier dès le début, mais isoler un petit morceau représentatif d'un pipeline

- Exemple : un script qui fait `read_csv → groupBy → mean.`
- Trouver ces bouts de code dans des notebooks Kaggle ou dépôts GitHub.

Passer dans un profiler

→ repérer hotspots

Un profiler est un outil qui mesure où ton code passe du temps et utilise de la mémoire.

Un hotspot (point chaud) = une partie du code qui consomme beaucoup de ressources (CPU, RAM, disque, réseau).

C'est souvent là qu'on peut optimiser ou paralléliser.

Vérifier si un LLM peut aussi suggérer ces hotspots

Au lieu de mesurer, demander à un LLM (comme ChatGPT) : “Voici mon code PySpark, selon toi quelles étapes risquent d'être lentes ou coûteuses ?”

Traduire vers Dask (semi-automatisé)

- Une fois qu'on sait quelle étape est lourde (ex: `groupBy`), l'écrire en Dask.
- Semi-automatisé = Définir des règles de correspondance entre Spark et Dask.
 - `withColumn` (Spark) → `assign` (Dask).
 - `groupBy` (Spark) → `groupby` (Dask).
 - `join` (Spark) → `merge` (Dask).

Plan

Résumé avec une analogie simple

Imaginons une voiture :

1. Extraction un bout de code → Prendre juste le moteur pour tester, mais pas sur toute la voiture
2. Profiler → Mettre un capteur qui mesure où ça chauffe le plus
3. Hotspot → Voir que c'est le moteur qui surchauffe, pas les pneus
4. LLM → un mécanicien qui regarde ton plan et dit : “attention, le moteur sera le point faible”
5. Traduire vers Dask → on change le moteur pour un plus léger (Dask)
6. Benchmarks → on mesure la vitesse avant/après



Merci.