TP : Spark Streaming

Objectifs

Ce TP a pour objectif de :

- S'initier au traitement de données en temps réel avec Apache Spark **DStream**
- Créer un flux de données en temps réel
- Appliquer des transformations simples

Environnement de travail

- Google Colab, Python 3, PySpark

Exercice 1 : Spark Streaming avec DStream

On souhaite analyser en temps réel des messages texte envoyés via un **socket** et calculer le **nombre d'occurrences de chaque mot** toutes les 5 secondes.

Travail demandé

1. Installer PySpark
2. Créer un serveur socket simulant un flux de données
3. Créer un `StreamingContext`
4. Lire les données sous forme de **DStream**
5. Appliquer un **WordCount streaming**
6. Afficher les résultats dans la console

```
!pip install pyspark

# Serveur socket simulant un flux
import socket, time, threading

def start_socket_server():
host = "localhost"
port = 9999
s = socket.socket()
s.bind((host, port))
s.listen(1)
conn, addr = s.accept()

messages = ["spark streaming dstream", "spark spark streaming", "big data
spark"]

while True:
for msg in messages:
```

```
conn.send((msg + "\n").encode())
time.sleep(2)

threading.Thread(target=start_socket_server, daemon=True).start()
```

```
from pyspark import SparkContext

from pyspark.streaming import StreamingContext

sc = SparkContext.getOrCreate()
ssc = StreamingContext(sc, 5)
lines = ssc.socketTextStream("localhost", 9999)
words = lines.flatMap(lambda line: line.split(" "))
pairs = words.map(lambda w: (w, 1))
counts = pairs.reduceByKey(lambda a, b: a + b)

counts.pprint()
ssc.start()
ssc.awaitTerminationOrTimeout(30)
ssc.stop(stopSparkContext=False)
```

Questions

1. Créer des sections pour chaque partie du travail demandé.
2. Chaque section doit être commentée
3. Modifier le code afin que la lecture aléatoire soit fait depuis un fichier
4. Qu'est-ce qu'un **micro-batch** dans DStream ?
5. Sur quelle structure repose un DStream ?
6. Quelle est la durée du batch utilisée