***Compte rendu :***

***BD NoSQL & Big Data***

***TP2 : Le traitement de batch avec le Hadoop streaming***

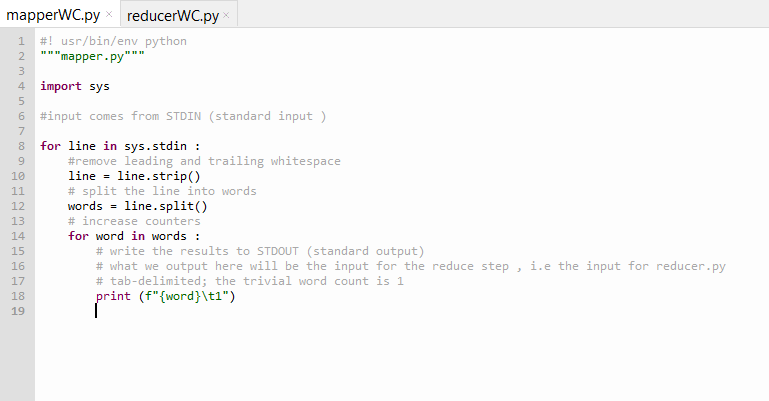
Réalisé par : Riahi Farah

Classe : ING2 INFO

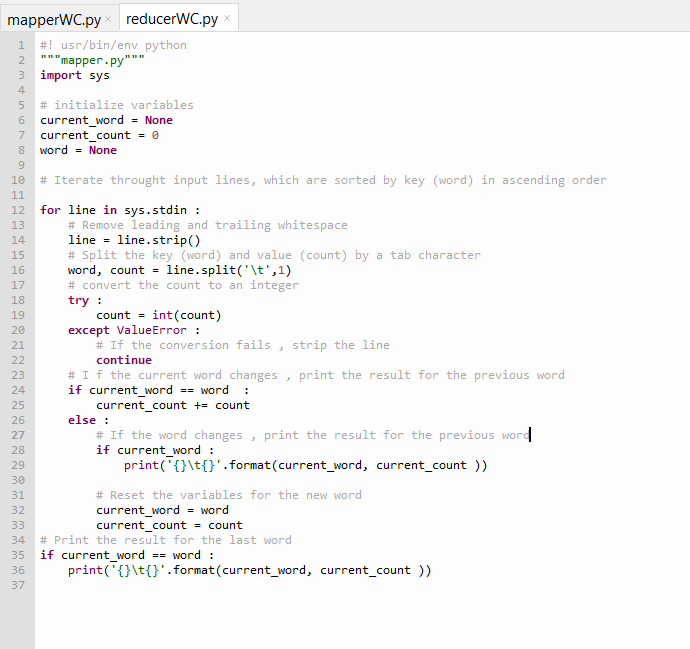
Objectifs du TP :

Ce TP vise à nous familiariser avec l'usage d'Hadoop Streaming pour la mise en œuvre de tâches de traitement de données via MapReduce en utilisant Python. Apprendre à développer des programmes de Map et de Reduce en Python, puis à exécuter des job MapReduce avec Hadoop Streaming

Partie 1 :

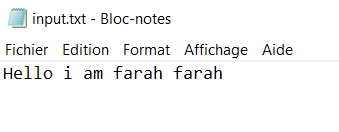


* Le code du Mapper est responsable de la transformation des données d'entrée en paires clé-valeur. Dans ce cas, chaque ligne est divisée en mots individuels. Pour chaque mot, le script émet une paire clé-valeur où la clé est le mot lui-même et la valeur est initialement définie sur 1. Ainsi, pour chaque occurrence d'un mot, le Mapper émet une paire clé-valeur où la clé est le mot et la valeur est 1.



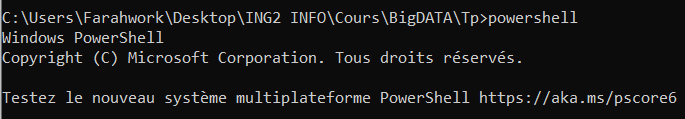
* Le code du Reducer est responsable de l'agrégation des paires clé-valeur générées par le Mapper. Il reçoit les paires clé-valeur triées par clé (mot) en entrée. Le Reducer somme les valeurs associées à chaque clé (mot) et produit une paire clé-valeur où la clé est le mot et la valeur est le nombre total d'occurrences de ce mot.

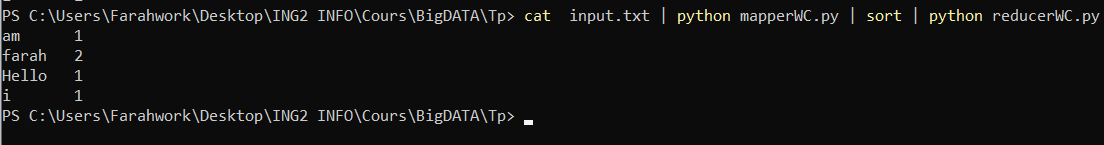
---------------------------------------------------------------------------------------------------------



* Fichier input.txt : utilisé comme exemple de données d'entrée pour tester le code du Mapper et du Reducer localement avant de l'exécuter sur un environnement Hadoop distribué.

Local :

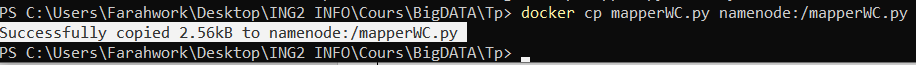




* lit le contenu du fichier input.txt, l'envoie au Mapper pour générer des paires clé-valeur, trie ces paires, puis les envoie au Reducer pour produire le résultat final du Word Count.

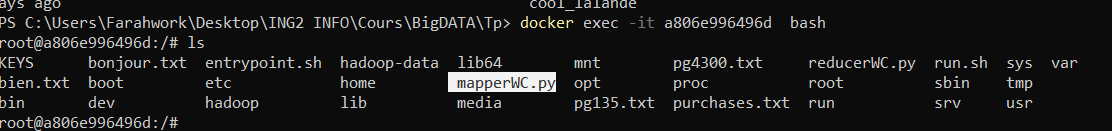
Exécution sur le cluster  :

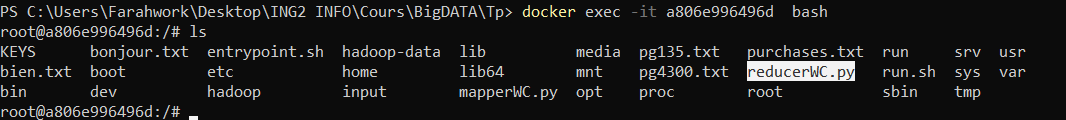
* Cette étape implique le transfert du code Mapper et Reducer ainsi que les données d'entrée vers le cluster Hadoop .



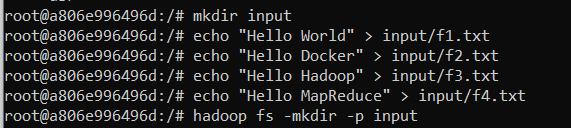


-------------------------------------------------------------------------------------------

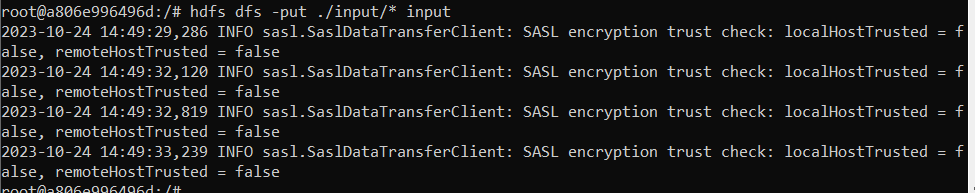




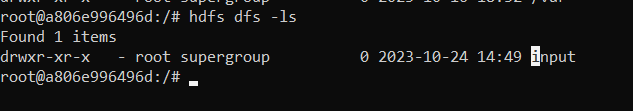
* Fichiers transférés dans le cluster Hadoop



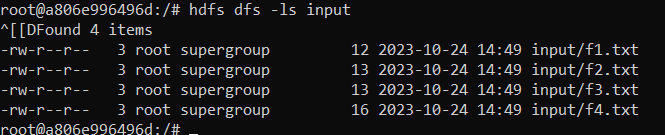
* Ces instructions créent des fichiers texte locaux : f1.txt, f2.txt, f3.txt, et f4.txt.
* Création du répertoire HDFS : Répertoire input dans HDFS à l'aide de la commande hadoop fs -mkdir -p input.



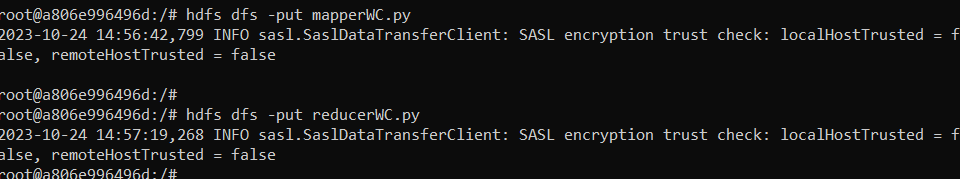
* Transférer les fichiers dans HDFS pour les rendre disponibles pour le traitement MapReduce sur le cluster Hadoop
* hdfs dfs -put ./input/ input : Cette commande est utilisée pour copier des fichiers ou des répertoires locaux vers HDFS. L'argument ./input/ fait référence au répertoire local qu’on souhaite transférer, tandis que l'argument final input spécifie le chemin dans HDFS où le contenu de ./input/ doit être placé.



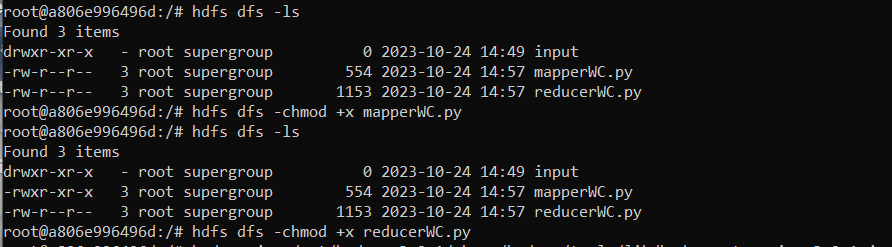
* le répertoire hdfs ‘’input’’ crée .



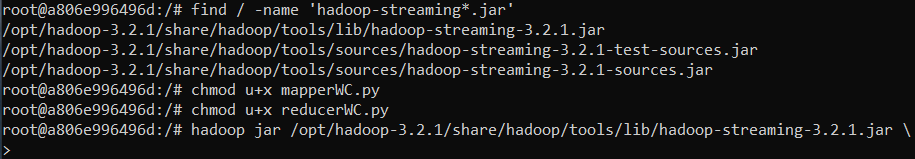
* fichiers texte transférés.



* transfert des scripts du Mapper et du Reducer sur le cluster Hadoop



* changement des droits : les fichiers sont exécutables.



* D'après la sortie de la commande find, le fichier hadoop-streaming-3.2.1.jar est situé dans le répertoire /opt/hadoop-3.2.1/share/hadoop/tools/lib/
* Le fichier JAR hadoop-streaming-3.2.1.jar est un composant important du framework Hadoop qui est utilisé pour prendre en charge les tâches de streaming dans Hadoop. Ce fichier JAR fait partie du module Hadoop Streaming, qui permet l'intégration de programmes MapReduce avec des langages de script non natifs, tels que Python.
* L’exécution de job a été faite et le résultat est dans output1

