Université Mohammed V de Rabat
Ecole Nationale Supérieure d'Arts et Métiers
Cycle Ingénieur Filière INDIA (3ème année) A.U. : 2023/2024 Pr. A. El Qadi

TP n1 en Big data Analytics

1. **Download**
   Hadoop 3.xxx (https://hadoop.apache.org/releases.html)
   Java 8.xxx
   Spark (https://spark.apache.org/downloads.html)
   Hadoop winutils (https://github.com/steveloughran/winutils)

   Décompresser winutils et copier le contenu du ..hadoop3..\bin dans le dossier de hadoop\bin
   c:\....\hadoop\bin

2. **Définir les variables d'environnement**
   Ajouter la variable d'environnement JAVA_HOME= C:\Program Files\Java\jdk1.8.0_66
   Ajouter la variable d'environnement HADOOP_HOME= …\hadoop-3.3.4
   Ajouter la variable d'environnement SPARK_HOME= …\spark-3.3.0-bin-hadoop3
   Modifier la variable d'environnement Path et rajouter :
   %JAVA_HOME%\bin
   %HADOOP_HOME%\bin
   %HADOOP_HOME%\sbin
   %SPARK_HOME%\bin

   Echo %PATH% pour vérifier
   A ce niveau vous pouvez tester par :

   Java –version

```
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
```

   Spark-shell --version

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.0
      /_/
Using Scala version 2.12.15, Java HotSpot(TM) 64-Bit Server VM, 1.8.0_66
Branch HEAD
Compiled by user ubuntu on 2022-06-09T19:58:58Z
Revision f74867bddfbcdd4d08076db36851e88b15e66556
Url https://github.com/apache/spark
Type --help for more information.
```

   Hdfs -version

```
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
```

   En cas d'erreur : JAVA_HOME is incorrectly set. Please update C:...\hadoop-2.7.1\conf\hadoop-env.cmd
   java environment path must not contain space. **The solution is as follows**:

   1. In the cmd line, charge the directory that contain the jdk (in my case C:\Program Files\Java\jdk1.8.0_73).
   2. execute the following line "for %I in (.) do echo %~sI" to display the short name of your installed jdk (in my case C:\PROGRA~1\Java\JDK18~1.0_7)
   3. in the file "hadoop-env.cmd", change the line "JAVA_HOME=%JAVA_HOME%" with "JAVA_HOME=C:\PROGRA~1\Java\JDK18~1.0_7".
   4. run again the file "hadoop-env.cmd" and it will work correctly.

3. **Lancer spark**
   spark-shell

Découvrir Apache Spark avec Scala.
Accéder à Spark à travers son shell, dans le répertoire de Spark : cd bin/ puis ./spark-shell

Exercice 1.
Écrire un programme Scala qui permet de lister tous les nombres impairs entre 10 et 100 (Fonction filter())

Exercice 2.
Supposons un panier d'achat avec les articles suivants : « Lait », « Fromage », « Beignets », « Pommes », « Bananes ».
1. Représenter les articles ci-dessus dans une structure de données appropriée,
2. Définir une fonction de valeur qui supprime tous les articles de fruits du panier. En d'autres termes, le panier ne doit contenir que les articles « Lait », « Fromage » et « Beignets ».

Exercice 3.
1. Créer un vecteur avec les éléments numériques suivants : 0, 10, 20, 47, -2, 99, -98.
2. Écrire un programme Scala pour trouver le plus petit et le plus grand élément numérique dans le vecteur.

Exercice 4.
1. Créer une structure de données appropriée pour représenter les valeurs numériques suivantes : 2, 8, 19, 20, 25, 50, 100, 10.
2. Définir une fonction de valeur qui identifiera si une valeur numérique donnée est divisible par 2.
3. Définir la fonction carre qui retourne le carré d'un élément (ici le paramètre de la fonction)
4. Utiliser la fonction map pour appliquer un traitement, calculer le carré de chaque élément de la liste en utilisant la fonction carre
5. Utiliser une fonction anonyme pour faire le même traitement
6. Définir une fonction Int→Boolean qui retourne true si un nombre est pair
7. Utiliser la fonction map pour transformer la liste des entiers (carre des éléments) en une liste de booleans
   Ex des resultat: res4: List[Boolean] = List(false, true, …)
8. Appliquer la fonction filter pour restituer que les éléments pairs de la liste
9. Appliquer la fonction foldLeft pour calculer la somme des éléments de la liste

Exercice 5.
1. Créer le fichier texte « input.txt »:
   « Spark est le projet open source, le plus actif dans le monde du Big Data.
   Il a commencé comme un projet de recherche à l'UC Berkeley AMPLab en 2009 et était open source au début de 2010. »
2. Charger ce fichier texte en utilisant l'API SparkContext
3. Séparer les mots selon les caractères d'espacement (produire plusieurs RDD à partir d'un RDD : flatMap)
4. Produire un RDD à partir d'un autre RDD : map, filter, reduceByKey
   4.1. Appliquer un map sur les mots obtenus pour produire le couple (<mot>,1)
   4.2. Appliquer un reduce qui permet de faire la somme des 1 des mots identiques
5. Persistance : transférer les résultats vers un fichier « file1out.count »