université PARIS-SACLAY | FACULTÉ DES SCIENCES D'ORSAY

lundi 28 novembre 2022

Traîtement Distribué des Données
Master 1 ISD

TP Scala & Spark

Exercice 0 Mise en place d'Hadoop et Spark

1. Utiliser les instructions du TP3 pour configurer Hadoop et y déposer les fichiers textes de ce TP :
   — si vous êtes sur une session de secours ou que vous avez fait le TP3 sur une session de secours, vous devez faire toutes les étapes préliminaires du TP 3
   — si vous avez fait le TP 3 sur votre compte, vous pouvez commencer à partir de l'étape 5 du TP3 (formattage du HDFS)
2. Créer un répertoire pour les fichiers de logs de spark :

   ```
   mkdir ~/.sparklogs
   ```

3. Démarer le serveur maître Spark :

   ```
   start-master.sh
   ```

   Visiter la page `http://localhost:9090` et noter l'URL du nœud maître (Par exemple `URL: spark://master-name:7077`). Lancer un nœud *worker* :

   ```
   start-worker.sh 'spark://master-name:7077'
   ```

   Recharger la page `http://localhost:9090` et vérifier qu'un worker est apparu dans le tableau des Workers.

Exercice 1

Utiliser un éditeur de texte (VS Code, gedit, geany, . . .) pour éditer les fichiers Python. L'utilisation d'eclipse n'est pas recommandée.

L'API de SPARK est documentée ici :

`https://spark.apache.org/docs/latest/api/python/reference/pyspark.html#rdd-apis`

1. Compléter le code du fichier `exo1.py` pour effectuer la transformation Map/Reduce word count du cours sur Hadoop, cette fois en utilisant l'API de SPARK. L'algorithme est similaire :
   — Charger le fichier (depuis HDFS) en un RDD spark qui représente la liste des lignes du fichiers
   — Séparer chaque ligne du fichier en utilisant une expression régulière. Est-il plus judicieux d'utiliser `.map()` ou `.flatMap()` ?
   — Associer à chaque mot de la liste le nombre 1
   — Utiliser enfin la méthode `.reduceByKey()` pour sommer tous les « 1 » associé à chacune des clés.
   — Sauver le RDD final dans un répertoire HDFS avec la méthode `.saveAsTextFile`
   Pour tester votre application, il suffit d'exécuter le code Python (`python3 exo1.py`)
2. Comparer l'approche Spark et l'approche Hadoop Map/Reduce.

Exercice 2

1. Créer dans un fichier `movie.py` une classe Python `Movie` de manière à ce que son constructeur accepte comme argument 5 chaînes de caractères (`id_s`, `title_s`, `year_s`, `runtime_s` et `rank_s` et initialise 4 attributs entiers (`id`, `year`, `runtime` et `rank`) et un attribut de type chaîne `title` avec les valeurs correspondantes des paramètres.

2. Redéfinir la méthode `__str__` pour qu'elle renvoie un film sous la forme de son titre, suivi de son année entre parenthèse.
3. Copier le fichier `exo1.py` en `exo2.py` et le modifier de manière à :
   — charger le fichier `movies.txt` depuis HDFS
   — Crée a partir du RDD de lignes un RDD de `Movie` (les lignes sont constitués dans l'ordre de l'id, du titre de l'année de la durée et du rang, séparés par des `;`).
   — Crée un RDD trié par années croissantes de `Movie`
   — Convertisse le RDD trié de `Movie` en RDD de chaînes (en appelant `str()` pour chaque movie)
   — Parcours le RDD de chaînes et l'affiche dans la console (on ne sauvera pas le résultat dans un fichier)

Exercice 3

Copier le fichier `exo2.py` en `exo3_1.py` (puis `exo3_2.py` etc. Vous remplacerez le nom de l'application et éventuellement les noms des fichiers d'entrée et de sortie). Implémenter les opérations suivantes :

1. Créer un RDD de `Movie`, ne gardez que les films sortis entre 1990 et 2000 et les afficher par ordre d'année décroissantes dans la sortie
2. Créer un RDD de `Movie`, compter pour chaque année le nombre de films et l'afficher par nombre de film croissant
3. Créer une classe `Role` et une classe `Person` similaires à `Movie`, charger trois RDD de `Role` (fichier `role.txt`), `Person` (fichier `people.txt` et `Movie` (fichier `movie.txt`) et donner pour chaque film la liste de ses acteurs avec leur roles (jointure entre les trois RDD sur l'id de film et l'id de personne). Dans le fichier `role.txt`, le premier champ est l'id du film et le second l'id de la personne).