Année universitaire : 2021-2022
Resp. module : M.-L. Messai

TD 3- TCSD

1 Objectifs

- Exécuter un premier traitement MapReduce (*job*).
- Ecrire un simple programme MapReduce en Python.
- Analyser le programme MapReduce écrit en Java.

2 Activité 1 : *Hello World*

- Lancez la VM Cloudera.
- Créez un fichier texte puis modifiez son contenu en ajoutant quelques phrases.
- Copiez le fichier sur HDFS.
- Lancez votre premier job MapReduce par la commande :

```
sudo -u cloudera hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount <votre fichier> output
```

- L'entrée du job est votre fichier. Le job compte les occurrences des mots dans votre fichier.
- Le résultat est enregistré dans le répertoire /user/cloudera/output/part-r-00000.
- Visualisez le contenu du résultat. Expliquez.

3 Activité 2 : *Word Count* sur des livres

- Récupérez les trois livres en fichiers texte sur Moodle et mettez-les dans un dossier nommé *books*. Il s'agit de « Les notes de Leonardo Da Vinci », « Les aventures de Sherlock Holmes » et « Les misérables ».
- Lancez un job MapReduce *wordcount* sur ces trois fichiers afin de compter les occurrences des mots.
- Consultez le résultat.

4 Activité 3 : Word Count en Python

Vous allez exécuter un programme qui imite le *wordcount*, c'est-à-dire qu'il lit les fichiers texte et compte la fréquence des mots. Le code Python utilise l'API Hadoop Streaming pour transmettre des données entre le *mapper* et le *reducer* via STDIN (entrée standard) et STDOUT (sortie standard). Nous utiliserons simplement le sys.stdin de Python pour lire les données d'entrée et imprimer notre propre sortie sur sys.stdout. C'est tout ce que nous devons faire car Hadoop Streaming s'occupe de tout le reste !

4.1 L'étape *map*

- Récupérez le code *mapper.py* sur Moodle et copiez-le sur la VM à /home/cloudera/mapper.py. Observez le code : Il lit les données de STDIN, les divise en mots et produit une liste de lignes mappant les mots à leurs comptes (intermédiaires) vers STDOUT. Le *mapper* ne calcule cependant pas une somme (intermédiaire) des occurrences d'un mot. Il produit immédiatement <mot> 1 tuples - même si un mot spécifique peut apparaître plusieurs fois dans l'entrée. Nous laissons l'étape de réduction suivante effectuer le décompte final.
- Testez votre scripts *mapper.py* localement :

```
echo "hadoop big data hadoop hdfs fs" | /home/cloudera/mapper.py
```

- Exécutez le job MapReduce avec seulement le *mapper* sur le dossier des livres du cluster Hadoop :

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.13.0.jar -D
mapreduce.job.reduce=0 -mapper /home/cloudera/mapper.py -input
/user/cloudera/books -output /user/cloudera/mapper-output
```

- -D mapreduce.job.reduce=0: permet de préciser qu'on ne veut pas d'étape *reduce*. Consultez le résultat /output_mapper sur HDFS.

4.2 L'étape *reduce*

- Récupérez le code *reducer.py* sur Moodle et copiez-le sur la VM à /home/cloudera/reducer.py. Observez le code : Il lit les résultats de *mapper.py* à partir de STDIN (donc le format de sortie de *mapper.py* et le format d'entrée attendu de *reducer.py* doivent correspondre) et additionne les occurrences de chaque mot à un compte final, puis affiche ses résultats dans STDOUT.
- Testez votre scripts *reducer.py* localement :

```
echo "hadoop big data hadoop hdfs fs" | /home/cloudera/mapper.py | sort -k1,1 |
/home/cloudera/reducer.py
```

- Exécutez le job Python MapReduce sur le cluster Hadoop. L'API Hadoop Streaming aide à transmettre des données entre le *mapper* et r*educer* via STDIN et STDOUT.

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.13.0.jar -
mapper /home/cloudera/mapper.py -reducer /home/cloudera/reducer.py -input
/user/cloudera/books -output /user/cloudera/python-output
```

- Consultez le résultat puis comparez-le avec le résultat de l'activité 2.
- Vous constatez que la ponctuation du texte pollue les résultats ? Proposez une amélioration du code pour que les caractères -,_ ![…etc ne soient pas compter.

5 Activité 3 : *Word Count* en Java (Optionnel)

- Récupérez le code en java de *Word Count* sur Internet.
- Compilez le code et créez son archive jar
- Lancez le job MapReduce dur le répertoire books.