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 n2 en Big Data Analytics

Configuration d'un seul cluster Hadoop à l'aide de docker-compose

1. Cloner le référentiel docker-Hadoop-spark **https://github.com/Marcel-Jan/docker-hadoop-spark.git**
   L'exemple de référentiel ci-dessus contient un ensemble Hadoop docker-compose.yml et est prêt à être déployé sur des conteneurs Docker.
   Démarrer le conteneur docker-Hadoop-spark à l'aide de la commande **docker-compose up –d**
   c:\Users\ADmiN\docker-hadoop-spark>docker-compose up -d
2. Une fois que tout est terminé, vérifier les conteneurs Hadoop en cours
   c:\Users\ADmiN\docker-hadoop-spark>docker container ls
   Dans la liste des conteneurs en cours d'exécution, obtenir un port pour le conteneur que vous souhaitez vérifier. Ces numéros de port sont déjà définis dans le référentiel précédemment cloné dans le fichier docker-compose.yml. Par exemple ;

Service	Identifiant	PORT
datanode	38e5ffb73407	9864
nodemanager	b4b78f0e3313	8042
namenode	c25780d97f98	9870
historyserver	a00f55f3203b	8188
resourcemanager	29493ebc0007	8088
spark-master	4c16f8b0b347	7077
spark-worker-1	45fdc3e57700	8081
hive-server	baf977faf87f	10000

   Et l'état actuel peut également être vérifié à l'aide de la page Web http://localhost:9870

Exercice 1.

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. Copier le fichier « input.txt » dans le conteneur namenode : **docker cp AP880212 namenode:/tmp/**
3. Exécuter le conteneur namenode de manière interactive : **docker exec -it namenode bash**
4. Dans le terminal bash résultant, créer le dossier d'entrée **/user/root/input**
5. Copier le fichier « input.txt » dans le dossier **/user/root/input** à l'aide de hdfs
6. Accéder au conteneur spark-master et exécuter-le de manière interactive

   C:\WINDOWS\system32>docker exec -it spark-master bash

7. Exécuter **spark-master**

8. Charger le dataset « input.txt » en utilisant l'API SparkContext

   scala> val lignes=spark.sparkContext.textFile("hdfs://namenode:9000/user/root/input/input.txt")

9. Création un RDD à partir de Hadoop datasets. Il est possible d'appliquer aux *Datasets* des actions, qui produisent des valeurs, et des transformations, qui produisent de nouveaux *Datasets.*

Actions :

1. Donner le nombre de lignes dans ce Dataset
2. Afficher la première ligne de ce Dataset
3. Afficher les 5 premières lignes de ce Dataset
4. Lister une partie du contenu de ce Dataset

Transformations et persistance

Les transformations peuvent :

produire un *Dataset* à partir d'un autre *Dataset* : map, filter, sort, etc.
produire un *Dataset* à partir de deux *Dataset* : union, join, crossJoin, etc.
produire 0 ou plusieurs *Dataset* à partir d'un *Dataset* : flatMap.

5. Séparer les mots selon les caractères d'espacement (produire plusieurs RDD à partir d'un RDD : flatMap)
6. Produire un RDD à partir d'un autre RDD : map, filter, reduceByKey
   a. Appliquer un map sur les mots obtenus pour produire le couple (<mot>,1)
   b. Appliquer un reduce qui permet de faire la somme des 1 des mots identiques
7. Persistance : transférer les résultats vers un fichier « file1out.count »

Exercice 2.

1. Charger le fichier AP880212
2. Afficher la structure de RDD
3. Compter le nombre de lignes du fichier à l'aide de l'action count.
4. Supprimer les lignes dont la longueur est nulle (length). Utiliser la transformation Spark filter.
5. A l'aide de la fonction Scala contains, compter le nombre de mot « <DOCNO> » dans le texte
6. On veut créer une nouvelle RDD qui va contenir le nombre de caractères par ligne. Donner la valeur de la ligne contenant le plus grand nombre de caractères, utiliser l'action reduce.
7. On veut écrire un wordCount en Spark sur le texte. A partir du résultat de la question 2, on veut compter le nombre d'occurrence de chaque mot dans le texte.
   7.a Créer une entrée dans une nouvelle RDD par mot, c'est-à-dire, qu'il faut découper les lignes par occurrences de " ". Pour cela, utiliser la transformation flatMap et la fonction split(" ").
   7b. A partir de la RDD obtenue en 7a, écrire le wordCount. Combien d'entrées avez-vous dans le résultat.
8. Trier la liste du résultat obtenu en question 7
   8a. Ordonner par mot (ordre alphabétique). Fournir le premier mot et le nombre d'occurrence.
   8b. Ordonner par nombre décroissant d'occurrences. Fournir le mot apparaissant le plus grand nombre de fois et son nombre d'occurrence.

Exercice 3.

1. Ecrire une courte séquence Scala qui affiche les nombres inférieurs à 10000 à la fois divisibles par 2 et par 15
   1.1 Générer les données
   1.2 Filtrer les données (garder seulement à la fois divisibles par 2 et par 15)

Version parallélisée (conteneur spark-master) :

2. Utiliser Spark-master pour distribuer les données et le filtrage de la question précédente sur le cluster Spark
   2.1 Générer les données
   2.2 Créer un RDD (paralelizedData) qui va contenir les données (distribuées sur les nœuds du cluster)
   2.3 Créer un RDD (filteredData) qui va contenir les données filtrées (distribuées sur les nœuds du cluster) (les données la fois divisible par 2 et par 15)
   2.4 Lancer l'action (collect) qui va déclencher le traitement parallèle et va retourner le résultat
   2.5 Suivre dans le UI l'exécution de votre script Spark http://localhost:4040
3. Ecrire une courte séquence Scala/Spark qui affiche la somme des nombres paires de 1 à 10000 (utiliser la fonction fold disponible sur un RDD qui est similaire a foldLeft de scala).
4. Ecrire une courte séquence Scala/Spark qui affiche le produit des nombres paires de 1 à 10000.
5. Combiner les deux derniers programmes en un seul. Mettre en cache les données au niveau des nœuds et vérifier la répartition de ces données via le Spark UI

Vous pouvez vérifier sur les nœuds le contenu du RDD mis en cache via le SparkUI (http://localhost:4040/storage/ puis cliquer sur le lien dans la colonne RDD Name)

6. Arrêter et supprimer les containers

docker-compose down