Hadoop et Map Reduce

Anthony Moisan

December 16, 2019

Contents

Préambule : ce projet est associé au cours Hadoop et Map_Reduce et se compose de deux projets pour implémenter Map Reduce et pour utiliser via Python et la librairie hdfs3 l'utilisation d'hadoop. Ce Notebook est autoportant avec des chemins en relatif avec l'arborescence des fichiers et dossiers associés.

1 Projet Map Reduce

1.1 Mapper et Reducer

1.1.1 Mapper

Le mapper va charger uniquement : * le premier champs et extraire le mois (deux premiers caractères) * le second champs pour ensuite filter les départs * le troisième champs pour connaitre la ville * le 11ème champs pour éviter le double comptage * le 13ème champs pour le maximum de seats

Les clés, valeurs exposés correspondent au mois et nombre de seats en filtrant sur la ville Sydney dans le cas d'un départ de cette ville et dans le cas où le champs stop est à 0.

1.1.2 Reducer

Le réducer utilise un dictionnaire permettant de stocker par mois les résultats en les sommant. Un test est réalisé pour s'assurer que le nombre de seats est un entier. On trie le dictionnaire de résultats pour l'affficher par mois croissant.

1.2 Test en local du mapper et reducer

[1]:
```
!cat Reservation/data.csv | python Reservation/Reservation_mapper.py | python␣
↪Reservation/Reservation_reduce.py
```

```
Month : 01       Number of passengers from Sydney : 6190421
Month : 02       Number of passengers from Sydney : 4766537
Month : 03       Number of passengers from Sydney : 6261788
Month : 04       Number of passengers from Sydney : 4960807
Month : 05       Number of passengers from Sydney : 4928827
Month : 06       Number of passengers from Sydney : 5935277
Month : 07       Number of passengers from Sydney : 5213361
Month : 08       Number of passengers from Sydney : 5112502
Month : 09       Number of passengers from Sydney : 6590224
Month : 10       Number of passengers from Sydney : 5187097
Month : 11       Number of passengers from Sydney : 5051415
Month : 12       Number of passengers from Sydney : 7250241
```

1.3 Execution HDFS

1.3.1 Creation répertoire HDFS

[2]:
```
!hadoop fs -mkdir /Reservation
```

1.3.2 Copie du fichier data.csv sur HDFS

[3]:
```
!hadoop fs -put Reservation/data.csv /Reservation/
```

[4]:
```
!hadoop fs -ls /Reservation/
```

```
Found 1 items
-rw-r--r--   1 cloudera supergroup    6126033 2019-12-16 01:25
/Reservation/data.csv
```

1.3.3 Lancement MAP Reduce

[5]:
```
!sudo hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/
 ↪hadoop-streaming-2.6.0-mr1-cdh5.13.0.jar \
-mapper "python $(pwd)/Reservation/Reservation_mapper.py" \
-reducer "python $(pwd)/Reservation/Reservation_reduce.py" \
-input /Reservation/data.csv \
-output /Reservation/out
```

```
packageJobJar: [] [/usr/lib/hadoop-mapreduce/hadoop-
streaming-2.6.0-cdh5.13.0.jar] /tmp/streamjob3915196115733774556.jar tmpDir=null
19/12/16 01:25:27 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
19/12/16 01:25:27 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
19/12/16 01:25:27 INFO mapred.FileInputFormat: Total input paths to process : 1
19/12/16 01:25:28 INFO mapreduce.JobSubmitter: number of splits:2
19/12/16 01:25:29 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1576155683072_0024
19/12/16 01:25:29 INFO impl.YarnClientImpl: Submitted application
application_1576155683072_0024
19/12/16 01:25:29 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1576155683072_0024/
19/12/16 01:25:29 INFO mapreduce.Job: Running job: job_1576155683072_0024
19/12/16 01:25:35 INFO mapreduce.Job: Job job_1576155683072_0024 running in uber
mode : false
19/12/16 01:25:35 INFO mapreduce.Job:  map 0% reduce 0%
19/12/16 01:25:43 INFO mapreduce.Job:  map 50% reduce 0%
19/12/16 01:25:44 INFO mapreduce.Job:  map 100% reduce 0%
19/12/16 01:25:50 INFO mapreduce.Job:  map 100% reduce 100%
19/12/16 01:25:50 INFO mapreduce.Job: Job job_1576155683072_0024 completed
```

```
successfully
19/12/16 01:25:50 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=88424
                FILE: Number of bytes written=614403
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=6130337
                HDFS: Number of bytes written=660
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=10859
                Total time spent by all reduces in occupied slots (ms)=3757
                Total time spent by all map tasks (ms)=10859
                Total time spent by all reduce tasks (ms)=3757
                Total vcore-milliseconds taken by all map tasks=10859
                Total vcore-milliseconds taken by all reduce tasks=3757
                Total megabyte-milliseconds taken by all map tasks=11119616
                Total megabyte-milliseconds taken by all reduce tasks=3847168
        Map-Reduce Framework
                Map input records=61065
                Map output records=8655
                Map output bytes=71108
                Map output materialized bytes=88430
                Input split bytes=208
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=88430
                Reduce input records=8655
                Reduce output records=12
                Spilled Records=17310
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=995
                CPU time spent (ms)=3710
                Physical memory (bytes) snapshot=912470016
                Virtual memory (bytes) snapshot=4690817024
                Total committed heap usage (bytes)=762839040
        Shuffle Errors
                BAD_ID=0
```

```
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=6130129
        File Output Format Counters
                Bytes Written=660
19/12/16 01:25:50 INFO streaming.StreamJob: Output directory: /Reservation/out
```

1.3.4 Recupération du fichier de résultats en local ou visualisation

[6]:
```
!hadoop fs -get /Reservation/out/part-00000
```

```
get: `part-00000': File exists
```

[7]:
```
!hadoop fs -cat /Reservation/out/part-00000
```

```
Month : 01       Number of passengers from Sydney : 6190421
Month : 02       Number of passengers from Sydney : 4766537
Month : 03       Number of passengers from Sydney : 6261788
Month : 04       Number of passengers from Sydney : 4960807
Month : 05       Number of passengers from Sydney : 4928827
Month : 06       Number of passengers from Sydney : 5935277
Month : 07       Number of passengers from Sydney : 5213361
Month : 08       Number of passengers from Sydney : 5112502
Month : 09       Number of passengers from Sydney : 6590224
Month : 10       Number of passengers from Sydney : 5187097
Month : 11       Number of passengers from Sydney : 5051415
Month : 12       Number of passengers from Sydney : 7250241
```

1.3.5 Supression de l'ensemble du répertoire et de son contenu

[8]:
```
!hadoop fs -rm -r /Reservation/
```

```
Deleted /Reservation
```

2 HDFS

Installation de la librairie hdfs3

[9]:
```
conda install -c conda-forge hdfs3
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

```
Note: you may need to restart the kernel to use updated packages.
```

2.1 Connexion à HDFS via hdfs3

[10]:
```
from hdfs3 import HDFileSystem
```

[11]:
```
hdfs = HDFileSystem(host="localhost", port=8020)
```

2.2 Création d'un répertoire sur HDFS

[12]:
```
hdfs.mkdir("/HDFS_Data/")
```

[13]:
```
hdfs.ls("/")
```

[13]:
```
['//HDFS_Data',
 '//Repertoire1',
 '//benchmarks',
 '//enterprise-deployment.json',
 '//hbase',
 '//solr',
 '//test',
 '//tmp',
 '//user',
 '//var']
```

2.3 Mettre les fichiers de données sur HDFS

[14]:
```
hdfs.put("HDFS_Data/data1.txt", "/HDFS_Data/data1.txt")
hdfs.put("HDFS_Data/data2.txt", "/HDFS_Data/data2.txt")
hdfs.put("HDFS_Data/data3.txt", "/HDFS_Data/data3.txt")
hdfs.put("HDFS_Data/data4.txt", "/HDFS_Data/data4.txt")
```

[15]:
```
hdfs.ls("/HDFS_Data/")
```

[15]:
```
['/HDFS_Data/data1.txt',
 '/HDFS_Data/data2.txt',
 '/HDFS_Data/data3.txt',
 '/HDFS_Data/data4.txt']
```

2.4 Comptabiliser le nombre de mots

[16]:
```
def count_words(file):
    word_counts = {}
    for line in file:
        for word in line.strip().split():
            if (word in word_counts) :
                word_counts[word] += 1
            else:
                word_counts[word] = 1
    return word_counts
```

Juste pour vérifier le fonctionnement de la fonction prédéfinie

[17]:
```
print(count_words(['Anthony Anthony Toto', 'Toto Titi']))
```

```
{'Anthony': 2, 'Toto': 2, 'Titi': 1}
```

On parcourt l'ensemble des fichiers hdfs, on les ouvre, on ajoute le contenu dans une liste puis on appelle la fonction pour déterminer le nombre de mots.

[18]:
```
import os
listFile = []
for file in hdfs.glob(os.path.join('/HDFS_Data', '*.txt')):
    with hdfs.open(file) as f:
        print(f.info())
        content = f.read()
        content = (str)(content)
        listFile.append(content)

resultWord = count_words(listFile)
```

```
{'kind': 'file', 'name': '/HDFS_Data/data1.txt', 'last_mod': 1576488372, 'size':
67742, 'replication': 3, 'block_size': 67108864, 'owner': 'cloudera', 'group':
'supergroup', 'permissions': 511, 'last_access': 1576488372, 'encryption_info':
None}
{'kind': 'file', 'name': '/HDFS_Data/data2.txt', 'last_mod': 1576488372, 'size':
68130, 'replication': 3, 'block_size': 67108864, 'owner': 'cloudera', 'group':
'supergroup', 'permissions': 511, 'last_access': 1576488372, 'encryption_info':
None}
{'kind': 'file', 'name': '/HDFS_Data/data3.txt', 'last_mod': 1576488372, 'size':
67933, 'replication': 3, 'block_size': 67108864, 'owner': 'cloudera', 'group':
'supergroup', 'permissions': 511, 'last_access': 1576488372, 'encryption_info':
None}
{'kind': 'file', 'name': '/HDFS_Data/data4.txt', 'last_mod': 1576488372, 'size':
67911, 'replication': 3, 'block_size': 67108864, 'owner': 'cloudera', 'group':
'supergroup', 'permissions': 511, 'last_access': 1576488372, 'encryption_info':
None}
```

[19]:
```
for key in sorted(resultWord) :
    print("%s %s"% (key, resultWord[key]))
```

```
' 4
Aenean 159
Aliquam 172
Aliquam. 1
Class 35
Cras 168
...
vulputate 142
vulputate, 11
vulputate. 28
vulputate.\r\n\r\nSed 1
```

[20]:
```
print("Le nombre de mots qui se repete : " + str(len(resultWord)))
```

```
Le nombre de mots qui se repete : 881
```

[21]:
```
print("Le top 10 des mots les plus repetes\n")
top10 = sorted(resultWord.items(), key=lambda k_v: k_v[1], reverse=True)[:10]
for (mot,count) in top10 :
    print("Le mot " + str(mot) + " est repete " + str(count))
```

```
Le top 10 des mots les plus repetes

Le mot et est repete 544
Le mot sit est repete 514
Le mot ac est repete 504
Le mot in est repete 477
Le mot sed est repete 452
Le mot id est repete 425
Le mot eget est repete 419
Le mot ut est repete 415
Le mot quis est repete 415
Le mot vel est repete 413
```

2.5 Suppression du répertoire HDFS_Data

[22]:
```
!hadoop fs -rm -r /HDFS_Data/
```

```
Deleted /HDFS_Data
```

[23]:
```
!hadoop fs -ls
```

```
Found 7 items
drwxr-xr-x   - cloudera cloudera          0 2019-12-12 13:39 Magasin
drwxr-xr-x   - cloudera cloudera          0 2019-12-13 02:02 SalaireMinMax
```

```
drwxr-xr-x   - cloudera cloudera          0 2019-12-12 14:46 WordCount
drwxr-xr-x   - cloudera cloudera          0 2019-12-12 13:17 countword
-rw-r--r--   1 cloudera cloudera      53655 2019-12-12 05:43 enterprise-
deployment.json
drwxr-xr-x   - cloudera cloudera          0 2019-12-12 13:00 temperature
drwxr-xr-x   - cloudera cloudera          0 2019-12-12 06:03 test
```