TP : Introduction à Map-Reduce

1.1 EXÉCUTION LOCALE

QUESTION 1 :
Map input records (données passées en entrée pour le Map) correspond aux nombre de lignes dans le fichier input.
Map output records (données passées en sortie pour le Map) correspond au nombre total de mots présents dans le fichier input.

QUESTION 2 :
Map output records est égal a Reduce input records car la sortie du Map est passée en entrée au Reduce. Le Reducer s'appuie sur le résultat en sortie du Mapper.

QUESTION 3 :
Correspond au nombre de mots distincts (sans redondances) présents dans le fichier input

1.2 PREMIER CONTACT AVEC HDFS

QUESTION 4 :
Chemin vers mon répertoire personnel depuis HDFS hdfs dfs -ls /user/mounaimc

1.4 EXÉCUTION SUR LE CLUSTER

QUESTION 5 :
Les fichiers en entrées sont séparés en petits morceaux (splits) afin d'être traités en parallèle par différents Mappers. Le nombre de splits correspond donc au nombre de morceaux de données.

1.5 COMBINER

QUESTION 6 :
Les compteurs qui permettent de vérifier que le combiner a fonctionné sont : Combine input records et Combine output records

QUESTION 7 :
Le compteur permet d'estimer le gain effectivement apporté par le combiner est **Map output materialized bytes**.
Il représente le nombre d'octets que le Map écrit en sortie. On remarque qu' en utilisant le combiner sur les 5 tomes des Misérables, le Map écrit en sortie 5 fois moins d'octets.
Sans combiner : Map output materialized bytes=5046030
Avec combiner : Map output materialized bytes=1148739

QUESTION 8 :

Le mot le plus répété sur les 5 Toms est : “de” avec 16757 occurrences.

```
 chouaib@mac0905  ~/Desktop/wordcount1_7  sort -r -n -k 2 part-r-00000 | head -10
de      16757
la      11025
et      10048
le      8471
à       7324
les     5278
un      4631
que     4288
il      4239
qui     4100
```

1.6 NOMBRE DE REDUCERS

QUESTION 9 :

Le compteur qui permet de refléter ce changement est **Launched reduce tasks**

QUESTION 10 :

La différence est que pour la part 1.5 on avait qu'un seul Reducer, donc un seul fichier de sortie avec le résultat du programme, tandis que là on utilise 3 Reducers qui génèrent chacun leur résultat dans un fichier à part, donc le résultat du programme est réparti sur 3 fichiers.

QUESTION 11 :

En testant sur les 5 Tomes des Misérables, on constate qu’avec 3 Reducer le temps d'exécution est multiplié par 3,8.

- Avec 1 Reducer :
  Total time spent by all map tasks (ms)=31502
  Total time spent by all reduce tasks (ms)=2828

- Avec 3 Reducer :
  Total time spent by all map tasks (ms)=33052
  Total time spent by all reduce tasks (ms)=10843

1.7 IN-MAPPER REDUCER

QUESTION 12 :

En analysant les figures 1, 2 et 3 ci-dessous, on constate que sans combiner (figure 1) le Map se déroule en approximativement 9 secondes, tandis qu’avec le combiner (figure 2) et in-mapper combiner (figure 3) 8 secondes suffisent.
En ce qui concerne le reducer, on constate que dans la figure 3 le Reduce dure 4 secondes, tandis que pour le combiner et in-mapper combiner (figures 2 et 3) il ne dure que 3 secondes.

```
mounaimc@im2ag-hadoop-01:~$ hadoop jar HadoopQ1_5.jar Question1_1 /data/miserables wordcount1_1
/data/miserables
2020-11-24 14:59:10,193 INFO client.RMProxy: Connecting to ResourceManager at /152.77.81.30:8032
2020-11-24 14:59:10,634 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/st
aging/mounaimc/.staging/job_1606181287639_0027
2020-11-24 14:59:10,884 INFO input.FileInputFormat: Total input files to process : 5
2020-11-24 14:59:11,062 INFO mapreduce.JobSubmitter: number of splits:5
2020-11-24 14:59:11,225 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1606181287639_0027
2020-11-24 14:59:11,226 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-11-24 14:59:11,387 INFO conf.Configuration: resource-types.xml not found
2020-11-24 14:59:11,387 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-11-24 14:59:11,443 INFO impl.YarnClientImpl: Submitted application application_1606181287639_0027
2020-11-24 14:59:11,477 INFO mapreduce.Job: The url to track the job: http://im2ag-hadoop-01.u-ga.fr:8088/proxy/ap
plication_1606181287639_0027/
2020-11-24 14:59:11,477 INFO mapreduce.Job: Running job: job_1606181287639_0027
2020-11-24 14:59:17,579 INFO mapreduce.Job: Job job_1606181287639_0027 running in uber mode : false
2020-11-24 14:59:17,580 INFO mapreduce.Job:  map 0% reduce 0%
2020-11-24 14:59:24,667 INFO mapreduce.Job:  map 20% reduce 0%
2020-11-24 14:59:25,674 INFO mapreduce.Job:  map 40% reduce 0%
2020-11-24 14:59:26,680 INFO mapreduce.Job:  map 100% reduce 0%
2020-11-24 14:59:30,700 INFO mapreduce.Job:  map 100% reduce 100%
2020-11-24 14:59:31,716 INFO mapreduce.Job: Job job_1606181287639_0027 completed successfully
2020-11-24 14:59:31,798 INFO mapreduce.Job: Counters: 54
```

Figure 1 : sans combiner (exercice 1.1)

```
mounaimc@im2ag-hadoop-01:~$ hadoop jar HadoopQ1_5.jar Question1_5 /data/miserables wordcount1_5
/data/miserables
2020-11-24 14:57:27,627 INFO client.RMProxy: Connecting to ResourceManager at /152.77.81.30:8032
2020-11-24 14:57:28,059 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/st
aging/mounaimc/.staging/job_1606181287639_0025
2020-11-24 14:57:28,316 INFO input.FileInputFormat: Total input files to process : 5
2020-11-24 14:57:28,458 INFO mapreduce.JobSubmitter: number of splits:5
2020-11-24 14:57:28,641 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1606181287639_0025
2020-11-24 14:57:28,642 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-11-24 14:57:28,825 INFO conf.Configuration: resource-types.xml not found
2020-11-24 14:57:28,826 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-11-24 14:57:28,881 INFO impl.YarnClientImpl: Submitted application application_1606181287639_0025
2020-11-24 14:57:28,927 INFO mapreduce.Job: The url to track the job: http://im2ag-hadoop-01.u-ga.fr:8088/proxy/ap
plication_1606181287639_0025/
2020-11-24 14:57:28,928 INFO mapreduce.Job: Running job: job_1606181287639_0025
2020-11-24 14:57:35,023 INFO mapreduce.Job: Job job_1606181287639_0025 running in uber mode : false
2020-11-24 14:57:35,024 INFO mapreduce.Job:  map 0% reduce 0%
2020-11-24 14:57:42,109 INFO mapreduce.Job:  map 40% reduce 0%
2020-11-24 14:57:43,116 INFO mapreduce.Job:  map 100% reduce 0%
2020-11-24 14:57:47,136 INFO mapreduce.Job:  map 100% reduce 100%
2020-11-24 14:57:47,144 INFO mapreduce.Job: Job job_1606181287639_0025 completed successfully
2020-11-24 14:57:47,229 INFO mapreduce.Job: Counters: 54
```

Figure 2 : Avec combiner (exercice 1.5)

```
mounaimc@im2ag-hadoop-01:~$ hadoop jar HadoopQ1_7.jar Question1_7 /data/miserables wordcount1_7
/data/miserables
2020-11-24 14:54:30,897 INFO client.RMProxy: Connecting to ResourceManager at /152.77.81.30:8032
2020-11-24 14:54:31,340 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/st
aging/mounaimc/.staging/job_1606181287639_0022
2020-11-24 14:54:31,594 INFO input.FileInputFormat: Total input files to process : 5
2020-11-24 14:54:32,148 INFO mapreduce.JobSubmitter: number of splits:5
2020-11-24 14:54:32,375 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1606181287639_0022
2020-11-24 14:54:32,376 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-11-24 14:54:32,536 INFO conf.Configuration: resource-types.xml not found
2020-11-24 14:54:32,536 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-11-24 14:54:32,591 INFO impl.YarnClientImpl: Submitted application application_1606181287639_0022
2020-11-24 14:54:32,629 INFO mapreduce.Job: The url to track the job: http://im2ag-hadoop-01.u-ga.fr:8088/proxy/ap
plication_1606181287639_0022/
2020-11-24 14:54:32,631 INFO mapreduce.Job: Running job: job_1606181287639_0022
2020-11-24 14:54:38,707 INFO mapreduce.Job: Job job_1606181287639_0022 running in uber mode : false
2020-11-24 14:54:38,708 INFO mapreduce.Job:  map 0% reduce 0%
2020-11-24 14:54:44,787 INFO mapreduce.Job:  map 20% reduce 0%
2020-11-24 14:54:45,795 INFO mapreduce.Job:  map 80% reduce 0%
2020-11-24 14:54:46,801 INFO mapreduce.Job:  map 100% reduce 0%
2020-11-24 14:54:49,818 INFO mapreduce.Job:  map 100% reduce 100%
2020-11-24 14:54:49,827 INFO mapreduce.Job: Job job_1606181287639_0022 completed successfully
2020-11-24 14:54:49,909 INFO mapreduce.Job: Counters: 54
```

Figure 3 : Avec in-mapper combiner (exercice 1.7)

Comme il n'y qu'un seul reducer qui s'exécute en parallèle (contrairement aux mappers), on peut se permettre de comparer les résultats des job Counters.
On constate une diminution considérable du temps d'exécution du reducer en utilisant le in-mapper combiner. D'ailleurs le temps d'exécution en parallèle décroit également.

Sans combiner : Exercice 1.1

```
Total time spent by all map tasks (ms)=27890
Total time spent by all reduce tasks (ms)=3500
```

Avec combiner : Exercice 1.5

```
Total time spent by all map tasks (ms)=27658
Total time spent by all reduce tasks (ms)=3061
```

Avec in-mapper combiner : Exercice 1.7

```
Total time spent by all map tasks (ms)=25073
Total time spent by all reduce tasks (ms)=2800
```

QUESTION 13 :

En analysant les figures 4, 5 et 6 ci-dessous on ne constate pas de grande différence concernant la quantité de mémoire utilisée.

```
Physical memory (bytes) snapshot=2000461824
Virtual memory (bytes) snapshot=15788974080
Total committed heap usage (bytes)=2094006272
Peak Map Physical memory (bytes)=366858240
Peak Map Virtual memory (bytes)=2640535552
Peak Reduce Physical memory (bytes)=249540608
Peak Reduce Virtual memory (bytes)=2631856128
```

Figure 4 : Sans combiner (exercice 1.1)

```
CPU time spent (ms)=10720
Physical memory (bytes) snapshot=2225102848
Virtual memory (bytes) snapshot=15794184192
Total committed heap usage (bytes)=2544369664
Peak Map Physical memory (bytes)=576409600
Peak Map Virtual memory (bytes)=2641821696
Peak Reduce Physical memory (bytes)=248131584
Peak Reduce Virtual memory (bytes)=2632527872
```

Figure 5 : Avec combiner (exercice 1.5)

```
CPU time spent (ms)=7780
Physical memory (bytes) snapshot=2032001024
Virtual memory (bytes) snapshot=15788331008
Total committed heap usage (bytes)=2029518848
Peak Map Physical memory (bytes)=364982272
Peak Map Virtual memory (bytes)=2637180928
Peak Reduce Physical memory (bytes)=247881728
Peak Reduce Virtual memory (bytes)=2633871360
```

Figure 6 : Avec in-mapper combiner (exercice 1.7)

2.1 MAP ET REDUCE

QUESTION 14 :

Ci-dessous le résultats des 5 tags les plus utilisés en utilisant comme fichier d'entrée flickrSampleSmall.txt

```
1 AG  3 الطوارق
2 AG  algeria 3
3 AG  amazigh culture 3
4 AG  3 تمنراست
5 AG  3 الهقار
6 BN  ghana 7
7 BN  lab 5
8 BN  africa 2
9 BN  idds 2
10 BN  single mothers 1
11 ML  mali 15
12 ML  niger 11
13 ML  islam 10
14 ML  viajes 10
15 ML  rio niger 10
16 UV  africa 10
17 UV  burkina-faso 9
18 UV  afrique 9
19 UV  burkina faso 9
20 UV  ghana 8
21
```

2.2 COMBINER

QUESTION 15 :

Pour pouvoir utiliser un combiner, le type de données intermédiaires de données doit être le suivant:

- Key : un **Text** correspondant au pays.
- Value : un tuple de **(Text,Integer)** correspondant à un tag et son nombre d'occurrences.
  On utilisera le type **StringAndInt** créé précédemment pour représenter ce tuple.
  Chaque tuple aura la valeur **1** comme nombre d'occurrence.

Mapper :	⇒	**Combiner :**	⇒	**Reducer :**
(Text , (Text, Integer))	⇒	(Text , (Text, Integer))	⇒	(Text , Text)

QUESTION 16 :

Voir la classe StringAndInt.java.

QUESTION 17 :

Voir la classe Question2_2.java.

Signature de la classe :

public static class **MyCombiner** extends **Reducer<Text, StringAndInt, Text, StringAndInt>**

QUESTION 18 :

Les tags les plus utilisés en france sont :

- **france** 35392
- **paris** 24254
- **barcelona** 12468
- **spain** 9779

```
354  FO  faroes 37
355  FO  torshavn 35
356  FR  france 35392
357  FR  paris 24254
358  FR  barcelona 12468
359  FR  spain 9779
360  FR  europe 6170
361  FS  nasa 1
362  FS  clouds 1
```

- **europe** 6170

QUESTION 19 :

Nous utilisons HashMap comme structure de données en mémoire dans le Reducer. Cette structure de données à l'avantage d'adapter dynamiquement sa taille au fur et à mesure qu'elle se remplie, on ne risque donc pas d'avoir des conflits ou débordements si le nombre de tags est très grand. Cela dit, on risque d'avoir une erreur du type **OutOfMemoryError.**

3. TOP-TAGS FLICKR PAR PAYS, AVEC MÉMOIRE LIMITÉE

QUESTION 20 :

- **Job 1 :** Le mapper lit en entrée un fichier texte et écrit en sortir un object **CountryAndTag** comme clé et **IntWritable** comme valeur.
  Le Combiner comme pour le Reducer, prennent en entrée un **CountryAndTag** comme clé et **IntWritable** comme valeur, et donnent en sortie la même chose.

  - **Mapper 1 : <LongWritable, Text, CountryAndTag, IntWritable>**
  - **Combiner 1 : <CountryAndTag, IntWritable, CountryAndTag, IntWritable>**
  - **Reducer 1 : <CountryAndTag, IntWritable, CountryAndTag, IntWritable>**

- **Job 2 :** Le mapper de ce job lit en entrée la sortie du reducer 1, c'est à dire un **CountryAndTag** comme clé et **IntWritable** comme valeur et écrit en sortie un **StringAndInt** et un Text.
  Le Reducer, prend en entrée un **StringAndInt** comme clé et un **Text** comme valeur, et écrit en sortie un **Text** et **StringAndInt**.

  - **Mapper 2 : <CountryAndTag, IntWritable, StringAndInt, Text>**
  - **Reducer 2 : <StringAndInt, Text, Text, StringAndInt>**

- **CountryComparator :** Classe qui hérite de writableComparator et qui implémente la méthode compare qui permet de comparer deux pays (fonction de découpage en groupes).

- **SortComparator :** Classe qui hérite de writableComparator et qui implémente la méthode compare qui permet de trier les tags (fonction de tri).

QUESTION 21 :

Voir **job 1** (MyMapper1, MyCombiner1, MyReducer1) dans le fichier Question3_1.java

QUESTION 22 :

Le second avantage de cette méthode pour la fonction reduce finale est qu'on a plus besoin d'utiliser de structure données (HashMap), on réduit ainsi la quantité de mémoire utilisée.

Les tags étant regroupés par pays et comptés dans la 1ere passe, et ensuite triés par ordre croissant par la fonction de tri, le Reduce final se contente donc d'écrire les K premiers éléments dans le fichier de sortie.

QUESTION 23 :

Voir **job 2** (MyMapper2, MyReducer2) dans le fichier Question3_1.java

QUESTION 24 :

S'il existe pour un même pays, plusieurs tags classés ex-aequo, le résultat sera toujours le même entre une exécution et une autre car le tri des tags se fait par ordre alphabétique.