TP 9: MAP REDUCE

Zakaria OUAKRIM

BDCC-2

Lien GitHub: https://github.com/ZakariaOuakrim/MapReduce-with-java

1- On souhaite développer un Job Map Reduce permettant, à partir d'un fichier texte (ventes.txt) en entré, contenant les ventes d'une entreprise dans les différentes villes, de déterminer le total des ventes par ville. La structure du fichier ventes.txt est de la forme suivante :

Date ville produit prix

Vous testez votre code en lançant le job sur le cluster Hadoop.

- **VentesMapper** : Cette classe MapReduce extrait la ville et le prix de chaque vente, puis envoie ces données pour un traitement ultérieur.

```java
// Mapper Class
public static class VentesMapper extends Mapper<Object, Text, Text, FloatWritable> {  1 usage  ZakariaOuakr

    private Text villeKey = new Text();  2 usages
    private FloatWritable prixValue = new FloatWritable();  2 usages

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

        String line = value.toString();
        String[] parts = line.split( regex: "\\s+");

        if (parts.length >= 4) {


            try {
                String ville = parts[1];
                float prix = Float.parseFloat(parts[3]);

                villeKey.set(ville);
                prixValue.set(prix);

                context.write(villeKey, prixValue);

            } catch (NumberFormatException e) {
                System.err.println("Invalid price format in line: " + line);
            }
        }
```

- **VentesReducer** : Cette classe MapReduce additionne les prix de chaque ville et retourne le total des ventes par ville en sortie.

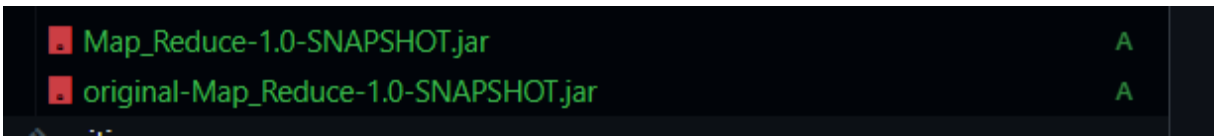
```java
public static class VentesReducer  2 usages  ZakariaOuakrim *
        extends Reducer<Text, FloatWritable, Text, FloatWritable> {

    private FloatWritable result = new FloatWritable();  2 usages

    public void reduce(Text key, Iterable<FloatWritable> values, Context context)
            throws IOException, InterruptedException {

        float sum = 0;
        for (FloatWritable val : values) {
            sum += val.get();
        }

        result.set(sum);

        context.write(key, result);
    }
}
```

- **Main :** Cette méthode configure et lance un job MapReduce qui calcule le total des ventes par ville à partir d'un fichier d'entrée.

```java
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "total ventes par ville");

    // Set jar
    job.setJarByClass(TotalVentesParVille.class);

    // Set Mapper and Reducer classes
    job.setMapperClass(VentesMapper.class);
    job.setReducerClass(VentesReducer.class);

    // Set Combiner class (same as Reducer for better performance)
    job.setCombinerClass(VentesReducer.class);

    // Set output types
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(FloatWritable.class);

    // Validate arguments
    if (args.length != 2) {
        System.err.println("Usage: TotalVentesParVille <input path> <output path>");
        System.exit(2);
    }

    // Set input and output paths
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    // Submit job and wait for completion
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

- **Générer le JAR**

```
@ZakariaOuakrim →/workspaces/MapReduce-with-java (master) $ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.zakaria:Map_Reduce >----------------------
```



- **Docker compose :** Ce fichier lance un cluster Hadoop complet avec HDFS, YARN, et un client, en connectant tous les services via un réseau commun.

```yaml
version: '3'

services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    restart: always
    ports:
      - 9870:9870
      - 9000:9000
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
    environment:
      - CLUSTER_NAME=hadoop-cluster
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    networks:
      - hadoop-network

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode
    restart: always
    volumes:
      - hadoop_datanode:/hadoop/dfs/data
    environment:
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    depends_on:
      - namenode
    networks:
      - hadoop-network

  resourcemanager:
    image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
    container_name: resourcemanager
```

- **Mes conteneurs :**

```
@ZakariaOuakrim ➜ /workspaces/MapReduce-with-java (master) $ docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED       STATUS                    PORTS                                                                                  NAMES
6979fb221edc   bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   4 hours ago   Up 34 minutes (healthy)   0.0.0.0:8188->8188/tcp, :::8188->8188/tcp                                              historyserver
decbb5495807   bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8     "/entrypoint.sh /run…"   4 hours ago   Up 13 minutes (healthy)   8042/tcp                                                                               nodemanager
50c4447e28f3   bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   4 hours ago   Up 34 minutes (healthy)   0.0.0.0:8088->8088/tcp, :::8088->8088/tcp                                              resourcemanager
199eef70394e   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8        "/entrypoint.sh /run…"   4 hours ago   Up 34 minutes (healthy)   9864/tcp                                                                               datanode
7eeb6232c237   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8        "/entrypoint.sh /run…"   4 hours ago   Up 34 minutes (healthy)   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9870->9870/tcp, :::9870->9870/tcp   namenode
@ZakariaOuakrim ➜ /workspaces/MapReduce-with-java (master) $
```

- Exemple du data

```
2023-05-13 Marseille Ordinateur 1299.99
2023-05-13 Paris Ecran 249.99
2023-05-14 Lyon Souris 24.99
2023-05-14 Paris Imprimante 199.99
2023-05-15 Marseille Tablette 349.99
2023-05-15 Lyon Camera 199.99
```

- Le conteneur essentiel Hadoop **Ressource Manager** :

```
@ZakariaOuakrim ➜ /workspaces/MapReduce-with-java (master) $ docker exec -it 50c4447e28f3 bash
root@50c4447e28f3:/#
```

- Copie le jar et le fichier du data :

```
root@50c4447e28f3:/# ls
KEYS  bin  boot  data.txt  dev  entrypoint.sh  etc  hadoop-data  home  lib  lib64  media  mnt  opt  original-Map_Reduce-1.0-SNAPSHOT.jar  proc  root  run  run.sh  sbin  srv  sys  tmp  usr  var
root@50c4447e28f3:/#
```

- Créer un dossier hdfs

```
root@50c4447e28f3:/# hdfs dfs -mkdir /input
```

- Copié les données vers le dossier :

```
root@50c4447e28f3:/# hdfs dfs -put data.txt /input/data.txt
```

- Lancer notre MapReduce Job

```
root@50c4447e28f3:/# yarn jar /original-Map_Reduce-1.0-SNAPSHOT.jar TotalVentesParVille /input/data.txt /result
2025-05-12 20:50:31,191 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-12 20:50:31,286 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-12 20:50:31,287 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-12 20:50:31,469 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool
2025-05-12 20:50:31,542 INFO input.FileInputFormat: Total input files to process : 1
2025-05-12 20:50:31,604 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-12 20:50:31,790 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local348789335_0001
2025-05-12 20:50:31,790 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-12 20:50:31,912 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-12 20:50:31,913 INFO mapreduce.Job: Running job: job_local348789335_0001
2025-05-12 20:50:31,914 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-12 20:50:31,927 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-12 20:50:31,927 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directo
2025-05-12 20:50:31,928 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-12 20:50:31,988 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-12 20:50:31,989 INFO mapred.LocalJobRunner: Starting task: attempt_local348789335_0001_m_000000_0
2025-05-12 20:50:32,014 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-12 20:50:32,014 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directo
2025-05-12 20:50:32,030 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
```

- Résultat :

```
root@50c4447e28f3:/# hdfs dfs -ls /
Found 4 items
drwxr-xr-x   - root supergroup          0 2025-05-12 16:47 /input
drwxr-xr-x   - root supergroup          0 2025-05-12 16:55 /o
drwxr-xr-x   - root supergroup          0 2025-05-12 16:54 /output
drwxr-xr-x   - root supergroup          0 2025-05-12 20:50 /result
```

```
root@50c4447e28f3:/# hdfs dfs -ls /result
Found 2 items
-rw-r--r--   3 root supergroup          0 2025-05-12 20:50 /result/_SUCCESS
-rw-r--r--   3 root supergroup         47 2025-05-12 20:50 /result/part-r-00000
root@50c4447e28f3:/#
```

```
root@50c4447e28f3:/# hdfs dfs -cat /result/part-r-00000
2025-05-12 20:52:30,319 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Lyon    854.47003
Marseille       2099.98
Paris   1709.95
root@50c4447e28f3:/#
```