



Amina Dahmouni

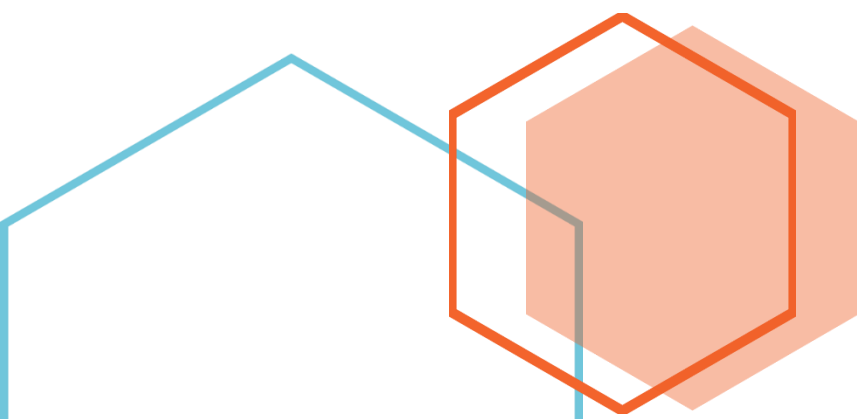
Classe : II-BDCC2

[Compte Rendu]

TP 3 : MapReduce et YARN

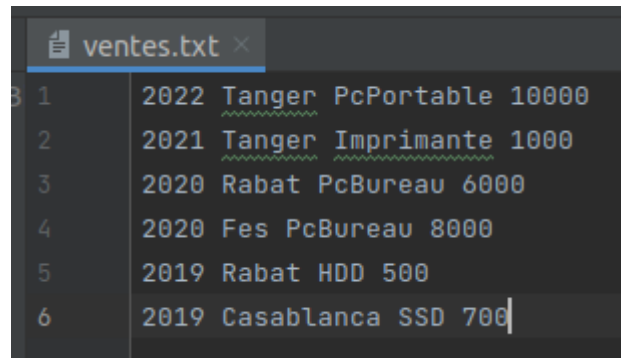
Big Data: Fondements et Architectures de stockage

**L'Ecole Normale Supérieure de l'Enseignement Technique de
Mohammedia (ENSET)**



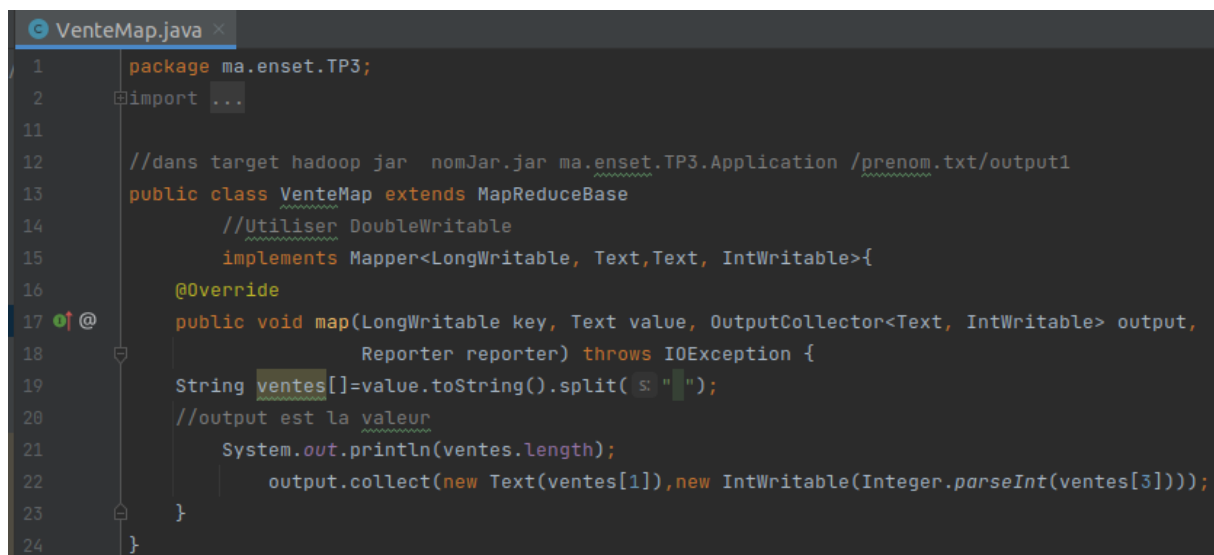
1. Déterminer le total des ventes par ville.

Le fichier text :



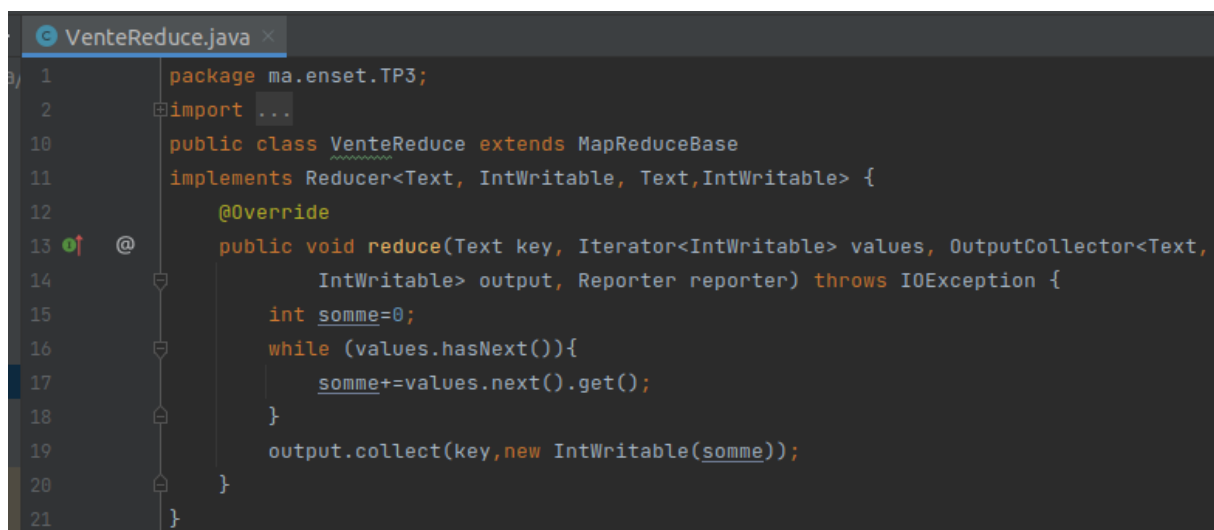
```
1 2022 Tanger PcPortable 10000
2 2021 Tanger Imprimante 1000
3 2020 Rabat PcBureau 6000
4 2020 Fes PcBureau 8000
5 2019 Rabat HDD 500
6 2019 Casablanca SSD 700
```

Map :



```
VenteMap.java
1 package ma.enset.TP3;
2 import ...
11
12 //dans target hadoop jar nomJar.jar ma.enset.TP3.Application /prenom.txt/output1
13 public class VenteMap extends MapReduceBase
14     //Utiliser DoubleWritable
15     implements Mapper<LongWritable, Text,Text, IntWritable>{
16
17     @Override
18     public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
19         Reporter reporter) throws IOException {
20         String ventes[]=value.toString().split(" ");
21         //output est la valeur
22         System.out.println(ventes.length);
23         output.collect(new Text(ventes[1]),new IntWritable(Integer.parseInt(ventes[3])));
24     }
25 }
```

Reduce :



```
VenteReduce.java
1 package ma.enset.TP3;
2 import ...
10 public class VenteReduce extends MapReduceBase
11     implements Reducer<Text, IntWritable, Text,IntWritable> {
12
13     @Override
14     public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text,
15         IntWritable> output, Reporter reporter) throws IOException {
16         int somme=0;
17         while (values.hasNext()){
18             somme+=values.next().get();
19         }
20         output.collect(key,new IntWritable(somme));
21     }
22 }
```

Application :

```
VenteReduce.java x MainApplication.java x
2
3 import ...
10 public class MainApplication {
11 @ public static void main(String[] args) throws IOException {
12     JobConf conf=new JobConf();
13     conf.setJobName("Total des ventes par villes");
14     conf.setJarByClass(MainApplication.class);
15
16     conf.setMapperClass(VenteMap.class);
17     conf.setReducerClass(VenteReduce.class);
18
19     conf.setOutputKeyClass(Text.class);
20     conf.setOutputValueClass(IntWritable.class);
21
22     conf.setInputFormat(TextInputFormat.class);
23     conf.setOutputFormat(TextOutputFormat.class);
24
25     FileInputFormat.addInputPath(conf,new Path("ventes.txt"));
26     FileOutputFormat.setOutputPath(conf,new Path("./output"));*/
27
28     FileInputFormat.addInputPath(conf,new Path(args[0]));
29     FileOutputFormat.setOutputPath(conf,new Path(args[1]));
30
31     JobClient.runJob(conf);
32 }
33 }
```

Résultat du traitement :

Browse Directory

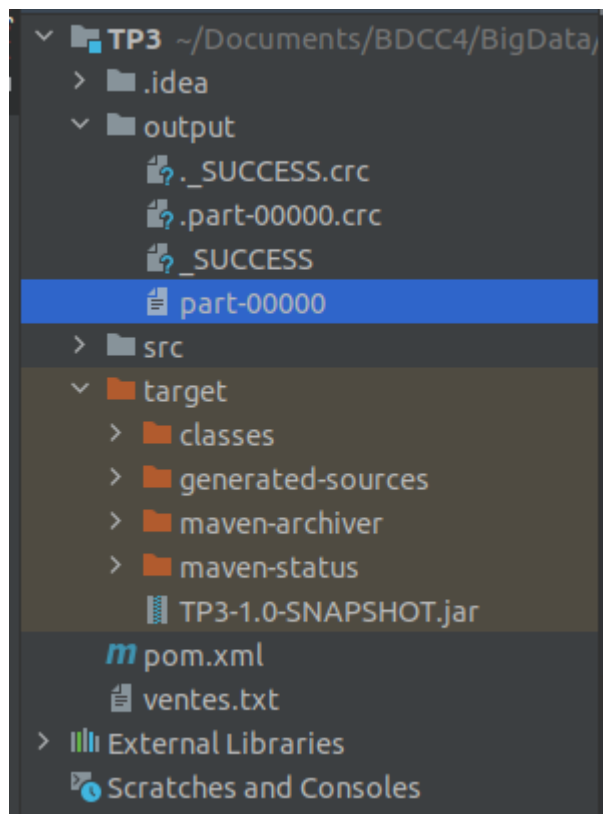
/output							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	3/16/2022, 5:34:32 PM	1	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	48 B	3/16/2022, 5:34:32 PM	1	128 MB	part-00000

Hadoop, 2015.

part-00000 (2) ~/Downloads		
1	Casablanca	700
2	Fes	8000
3	Rabat	6500
4	Tanger	11000

Dans le local :

```
FileInputFormat.addInputPath(conf,new Path( pathString: "ventes.txt"));
FileOutputFormat.setOutputPath(conf,new Path( pathString: "./output"));
```



part-00000 x		
1	Casablanca	700
2	Fes	8000
3	Rabat	6500
4	Tanger	11000
5		

2. Calculer le prix total des ventes des produits par ville pour une année donnée.

Map :

```
Q2_Map.java x
1 package ma.enset.TP3;
2 import ...
10 public class Q2_Map extends MapReduceBase
11     implements Mapper<LongWritable, Text, Text, IntWritable> {
12
13     @Override
14     public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
15                     Reporter reporter) throws IOException {
16         //Key double
17         String year_city=value.toString().split(" ")[0]+" "+value.toString().split(" ")[1];
18         System.out.println(year_city);
19         String ventes[]=value.toString().split(" ");
20         System.out.println(ventes.length);
21         output.collect(new Text(year_city), new IntWritable(Integer.parseInt(ventes[3])));
22     }
23 }
```

On génère un clé double qui dépend de l'année et de la ville au même temps « year_city ».

Reduce :

```
Q2_Reduce.java x
1 package ma.enset.TP3;
2 import ...
10
11 public class Q2_Reduce extends MapReduceBase
12     implements Reducer<Text, IntWritable, Text,IntWritable> {
13     @Override
14     public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text,
15                         IntWritable> output, Reporter reporter) throws IOException {
16         int somme=0;
17         while (values.hasNext()){
18             somme+=values.next().get();
19         }
20         output.collect(key,new IntWritable(somme));
21     }
22 }
```

Application.java :

```
Q2_Application.java x
1 package ma.enset.TP3;
2 import ...
7 public class Q2_Application {
8     public static void main(String[] args) throws IOException {
9         JobConf conf=new JobConf();
10        conf.setJobName("Total des ventes par villes pour une année donnée");
11        conf.setJarByClass(Q2_Application.class);
12
13        conf.setMapperClass(Q2_Map.class);
14        conf.setReducerClass(Q2_Reduce.class);
15
16        conf.setOutputKeyClass(Text.class);
17        conf.setOutputValueClass(IntWritable.class);
18
19        conf.setInputFormat(TextInputFormat.class);
20        conf.setOutputFormat(TextOutputFormat.class);
21
22        FileInputFormat.addInputPath(conf,new Path( pathString: "ventes.txt"));
23        FileOutputFormat.setOutputPath(conf,new Path( pathString: "./output1"));
24
25        /*FileInputFormat.addInputPath(conf,new Path(args[0]));
26        FileOutputFormat.setOutputPath(conf,new Path(args[1]));*/
27
28        JobClient.runJob(conf);
29    }
30 }
```

Résultat :

Browse Directory

/							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	3/2/2022, 11:30:51 PM	0	0 B	BDCC
drwxr-xr-x	root	supergroup	0 B	3/16/2022, 5:34:32 PM	0	0 B	output
drwxr-xr-x	root	supergroup	0 B	3/19/2022, 7:09:03 PM	0	0 B	output1
-rw-r--r--	root	supergroup	75 B	3/15/2022, 10:14:24 AM	1	128 MB	prenom.txt
-rw-r--r--	root	supergroup	142 B	3/19/2022, 7:04:27 PM	1	128 MB	ventes.txt

Hadoop, 2015.

part-00000 (3) ~/Downloads			
Open	▼	+	
1	2019	Rabat	1200
2	2020	Fes	8000
3	2022	Rabat	6000
4	2022	Tanger	11000