

RAPPORT DU TP3

MapReduce et YARN



Abderrahmane ET-Tounani

IIBDCC-2

2022-2023

Objectifs du TP

Initiation au framework hadoop et au patron MapReduce

Outils et Versions

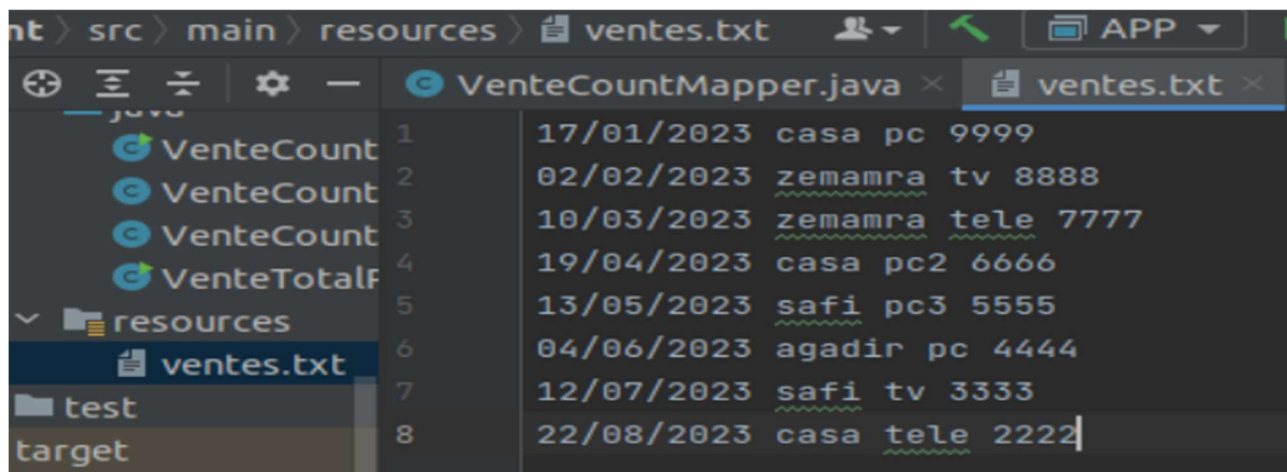
- [Apache Hadoop](#) Version: 2.7.2.
- [IntelliJ IDEA](#) Version Ultimate 2016.1
- [Java](#) Version 1.8.

Présentation

Pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-common</artifactId>
    <version>2.7.2</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapred
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-mapreduce-client-core</artifactId>
    <version>2.7.2</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-hdfs -
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-hdfs</artifactId>
    <version>2.7.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-mapreduce-client-common</artifactId>
    <version>2.7.2</version>
  </dependency>
</dependencies>
```

Fichier texte « ventes.txt ». la forme du fichier est : **Date ville produit prix.**



1- Question 1

Driver

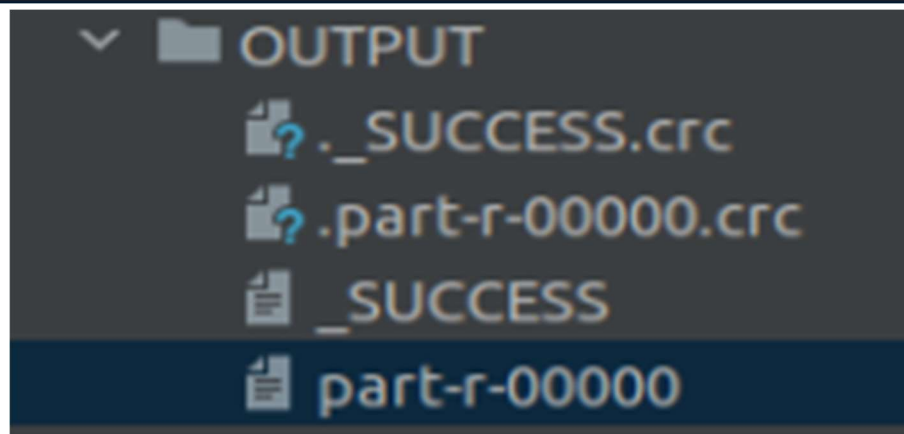
```
1 public class VentesCount {
2     public static void main(String[] args) throws Exception {
3         Configuration conf = new Configuration();
4         Job job = Job.getInstance(conf, "Vente count");
5         job.setJarByClass(WordCount.class);
6         job.setMapperClass(TokenizerMapper.class);
7         job.setCombinerClass(IntSumReducer.class);
8         job.setReducerClass(IntSumReducer.class);
9         job.setOutputKeyClass(Text.class);
10        job.setOutputValueClass(IntWritable.class);
11        FileInputFormat.addInputPath(job, new Path(args[0]));
12        FileOutputFormat.setOutputPath(job, new Path(args[1]));
13        System.exit(job.waitForCompletion(true) ? 0 : 1);
14    }
15 }
```

Reducer

```
1 public class IntSumReducer
2     extends Reducer<Text,IntWritable,Text,IntWritable> {
3     private IntWritable result = new IntWritable();
4     public void reduce(Text key, Iterable<IntWritable> values,
5         Context context
6     ) throws IOException, InterruptedException {
7         int sum = 0;
8         for (IntWritable val : values) {
9             System.out.println("Ville : "+val.get());
10            sum += val.get();
11        }
12        System.out.println("--> Sum Ventes = "+sum);
13        result.set(sum);
14        context.write(key, result);
15    }
16 }
```

Mapper

```
1 public class VenteCountMapper extends Mapper<LongWritable,  
    Text,Text,IntWritable> {  
2     @Override  
3     protected void map(LongWritable key, Text value, Mapper<LongWritable,  
        Text, Text, IntWritable>.Context context) throws IOException,  
        InterruptedException {  
4         String ventes[] = value.toString().split(" ");  
5         context.write(new Text(ventes[1]), new IntWritable(1));  
6     }  
7 }
```



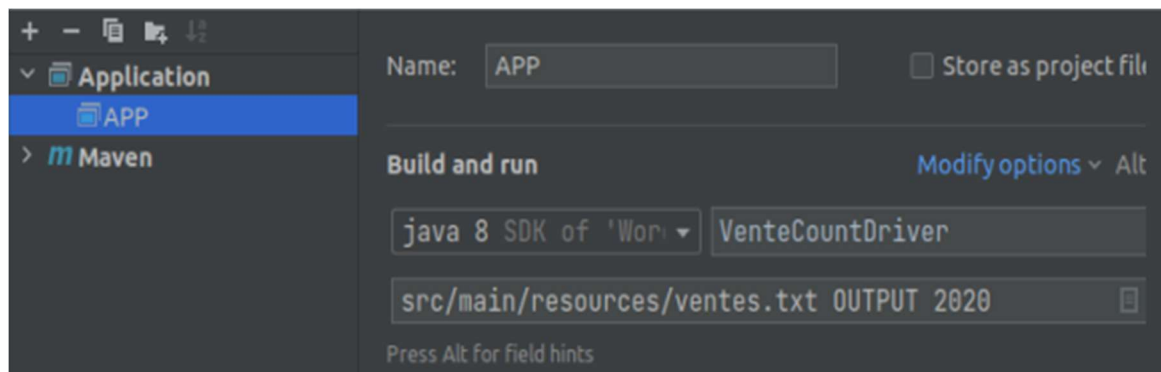
agadir	1
casa	3
safi	2
zemamra	2

2- Question 2

```
1 public static void main(String[] args) throws Exception {
2     Configuration conf = new Configuration();
3     conf.set("year", args[2]); // set total sales by year argument in
    configuration
4     Job job = Job.getInstance(conf, "Vente count");
5     job.setJarByClass(WordCount.class);
6     job.setMapperClass(TokenizerMapper.class);
7     job.setCombinerClass(IntSumReducer.class);
8     job.setReducerClass(IntSumReducer.class);
9     job.setOutputKeyClass(Text.class);
10    job.setOutputValueClass(IntWritable.class);
11    FileInputFormat.addInputPath(job, new Path(args[0]));
12    FileOutputFormat.setOutputPath(job, new Path(args[1]));
13    System.exit(job.waitForCompletion(true) ? 0 : 1);
14 }
```

```
1 public class VenteCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
2     @Override
3     protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
        IntWritable>.Context context) throws IOException, InterruptedException {
4         String ventes[] = value.toString().split(" ");
5         Configuration conf = context.getConfiguration();
6         String year = conf.get("year").toString();
7         if(ventes[0].contains(year))
8             context.write(new Text(ventes[1]), new IntWritable(Integer.parseInt(ventes[3])));
9     }
10 }
```

Pour le test en local



Pour le test en hadoop

`hadoop jar <path_to_jar_file> <main_class> <input_path> <output_path> <year>`