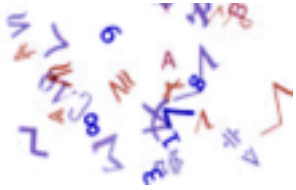


TP Map-Reduce

*Travail à rendre par mail
avant le lundi 1^{er} octobre*



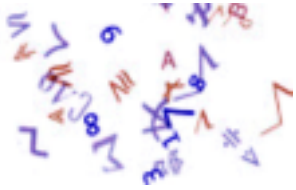


*Expérimentez le fonctionnement de l'exemple
donnée en cours*

Calculer la fréquence des mots présents
dans les ouvrages de l'ABU.

mot (nombre d'occurrences)





Mapper

Java Programming

```
public static class WordCountMapper
    extends Mapper< Object, Text, Text, IntWritable >
{

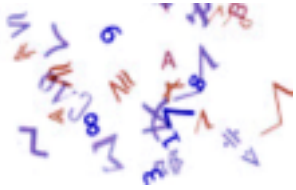
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

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

        StringTokenizer itr = new StringTokenizer(
            value.toString().toLowerCase(),
            "\t\n\r\f.,;:-'\"\\()!/?/[{}]«»#=/+`*@&$"
        );

        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}
```





Reducer

Java Programming

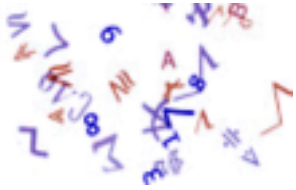
```
public static class WordCountReducer
    extends Reducer< Text, IntWritable, Text, IntWritable >
{
    private IntWritable result = new IntWritable ();

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException
    {
        int sum = 0;

        for (IntWritable val : values) {
            sum += val.get();
        }

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

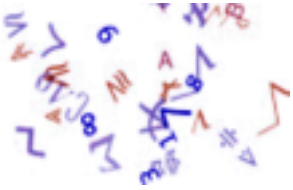




Java Programming

```
public class WordCount {  
    public static void main(String[] args) throws Exception {  
        Configuration conf = new Configuration();  
        Job job = Job.getInstance(conf, "Word count");  
  
        job.setJarByClass(WordCount.class);  
        job.setMapperClass(WordCountMapper.class);  
        job.setReducerClass(WordCountReducer.class);  
  
        job.setOutputKeyClass(Text.class);  
        job.setOutputValueClass(IntWritable.class);  
  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        FileOutputFormat.setOutputPath(job, new Path(args[1]));  
  
        System.exit(job.waitForCompletion(true) ? 0 : 1);  
    }  
}
```





Java Programming

◆ Compile

```
hadoop com.sun.tools.javac.Main WordCount.java  
jar cf WC.jar WordCount*.class
```

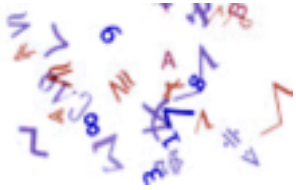
◆ Execute

```
hadoop jar WC.jar WordCount input output
```

◆ Results

```
hdfs dfs -ls output  
hdfs dfs -cat 'output/part-*
```





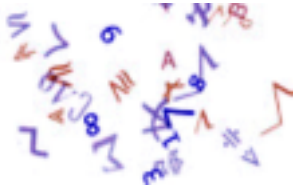
Exercice

Calculer l'index des mots présents
dans les ouvrages de l'ABU.

`mot (liste des ouvrages contenant ce mot)`

TP a rendre à la fin de la séance par mail
adressé à jaillon@emse.fr et ayant pour
sujet : [TP BIGDATA] <votre nom>





File containing this split's data.

```
import org.apache.hadoop.mapreduce.lib.input.FileSplit;

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

    FileSplit fileSplit = (FileSplit)context.getInputSplit();
    String filename = fileSplit.getPath().getName();

    ...
}
```

Documentation: <hadoop_dir>/share/doc/hadoop/api/index.html

