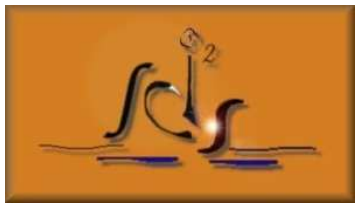
Ciencia de Datos: Un Enfoque Práctico en la Era del Big Data
Hadoop: Caso Práctico 1

Sara Del Río García

Departamento de Ciencias de la Computación e Inteligencia Artificial,
E.T.S. de Ingenierías Informática y de Telecomunicación,
srio@decsai.ugr.es



Contenido

Contenido

Ejemplo: WordCount

- Tenemos un fichero de gran tamaño que contiene secuencias de palabras

- **OBJETIVO**:
  Contar el número de veces que aparece cada palabra en el fichero

Contenido

WordCount usando MapReduce

Pseudocódigo

Algorithm 1 map(key, value) // key: document ID; value: text of document:

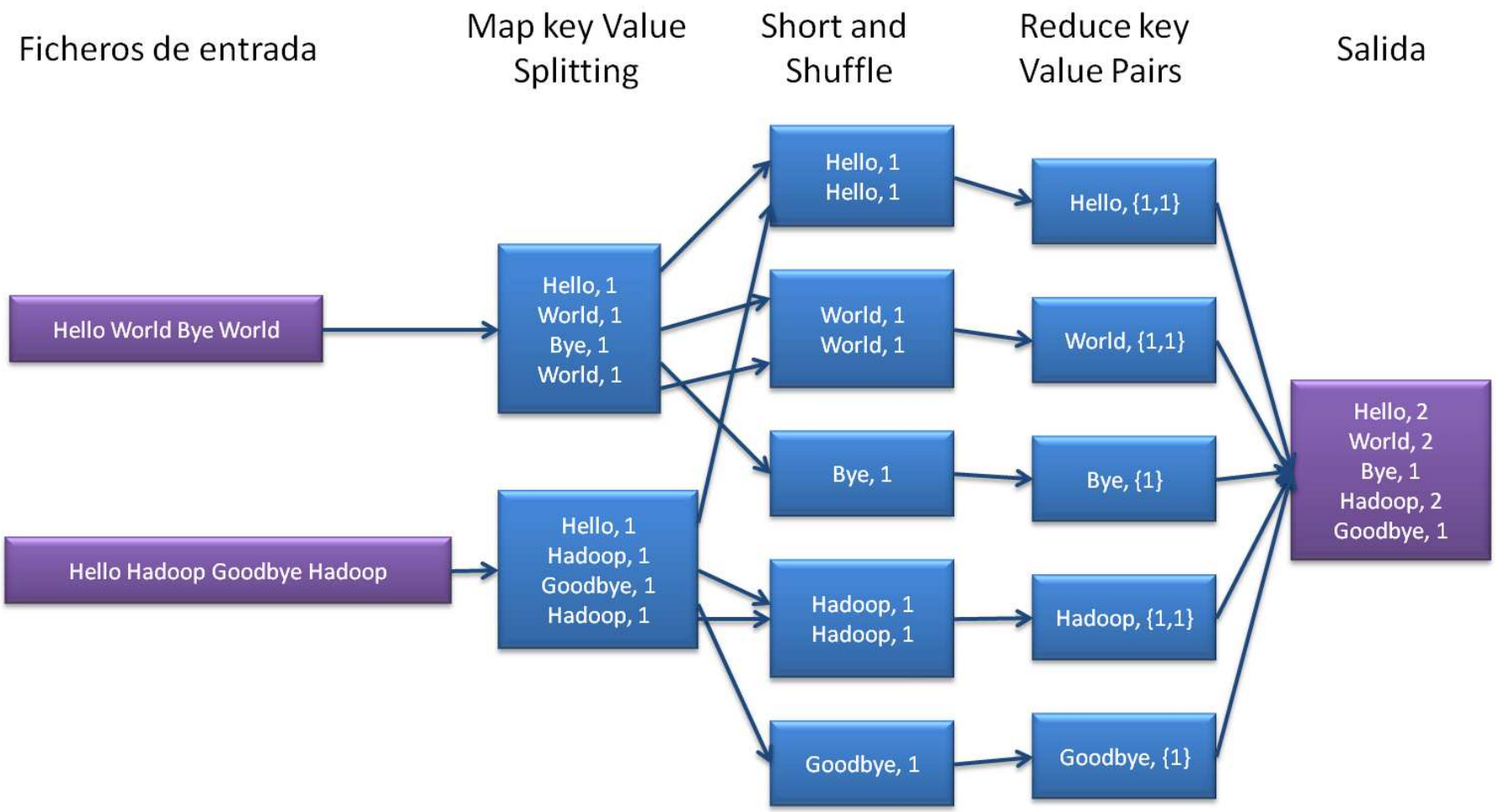
```
1: for each word w in value do
2:     EMIT (w, 1)
3: end for
```

Algorithm 2 reduce(key, value-list) // key: a word; value-list: a list of integers (theoretically// 1, but ... "combiners")

```
1: result ← 0
2: for each count v on value − list do
3:     result ← result + v
4: end for
5: EMIT (key, result)
```

WordCount usando MapReduce

Walkthrough of WordCount example (but ... "combiners").



WordCount usando MapReduce

Main Class

```
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WordCount {
```

WordCount usando MapReduce

Mapper Class

```
        public static class Map extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, IntWritable>
              {

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

  public void map(LongWritable key, Text value,
      OutputCollector<Text, IntWritable> output, Reporter
      reporter) throws IOException {

    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
      word.set(tokenizer.nextToken());
      output.collect(word, one);
    }
  }
}
```

WordCount usando MapReduce

Reducer Class

```
        public static class Reduce extends MapReduceBase
        implements Reducer<Text, IntWritable, Text, IntWritable>
              {

    public void reduce(Text key, Iterator<IntWritable> values,
         OutputCollector<Text, IntWritable> output, Reporter
         reporter) throws IOException {

      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
}
```

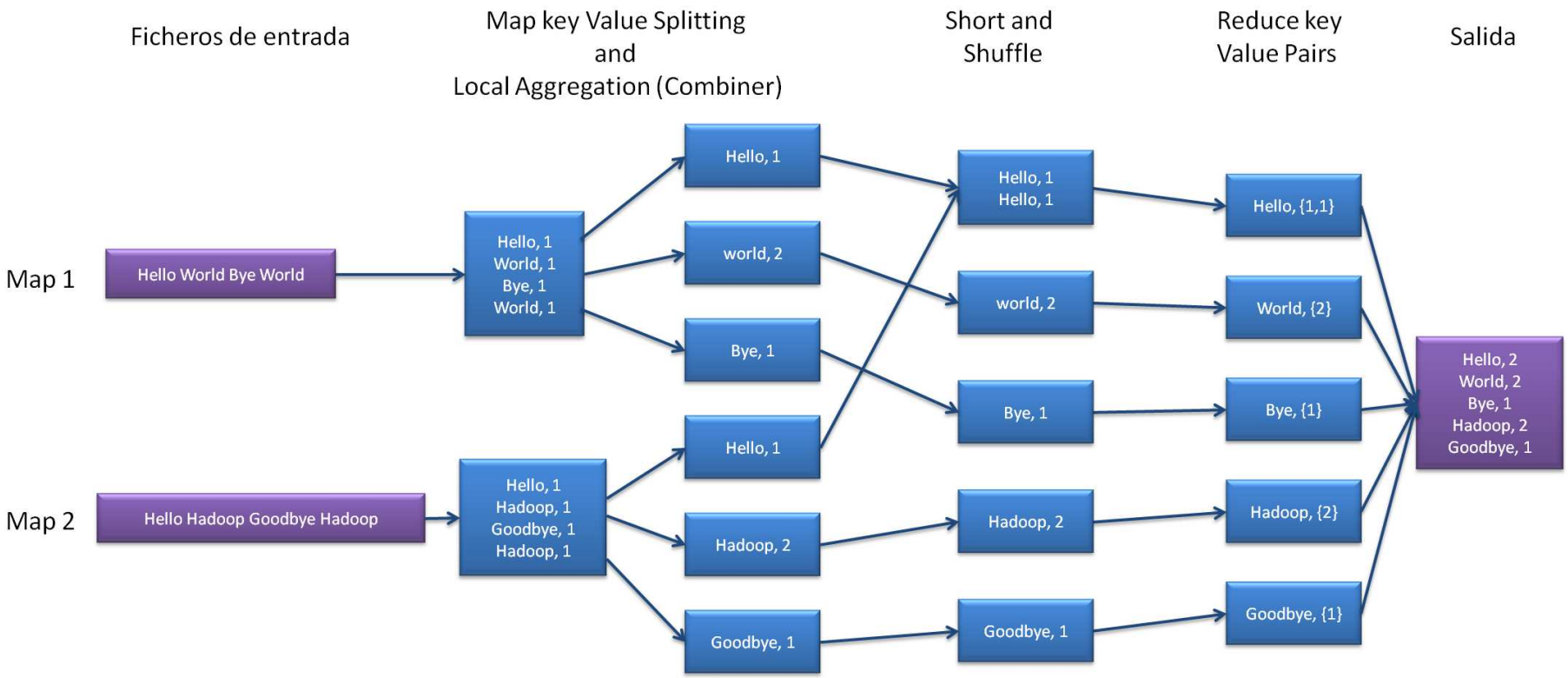
WordCount usando MapReduce

Main method

```
        public static void main(String[] args) throws Exception
            {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("wordcount");

    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);

    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
  }
}
```

WordCount usando MapReduce

Walkthrough of WordCount example (with "combiners").



WordCount usando MapReduce

- Uso:
  1. Crear el directorio de entrada "/user/hadoop/wordcount/input" en HDFS:

     hadoop fs -mkdir /user/hadoop/wordcount
     /user/hadoop/wordcount/input

  2. Crear ficheros de texto de ejemplo y copiarlos al directorio creado previamente en HDFS:

     echo "Hello World Bye World" > file0

     echo "Hello Hadoop Goodbye Hadoop" > file1

     hadoop fs -put file* /user/hadoop/wordcount/input

WordCount usando MapReduce

- Uso:
  3. Compilar "WordCount.java":

     mkdir wordcount_classes

     javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d wordcount_classes WordCount.java

  4. Crear el JAR:

     jar -cvf wordcount.jar -C wordcount_classes/ .

  5. Ejecutar la aplicación:

     hadoop jar wordcount.jar org.myorg.WordCount /user/hadoop/wordcount/input /user/hadoop/wordcount/output

WordCount usando MapReduce

- Uso:
  6. Comprobar la salida:

     hadoop fs -cat wordcount/output/part-00000

     Obtendermos la siguiente salida:

     Bye 1
     Goodbye 1
     Hadoop 2
     Hello 2
     World 2

Contenido

Referencias

- Hadoop Tutorial - WordCount:

  http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht_wordcount1.html

Happy Hadooping!

Ciencia de Datos: Un Enfoque Práctico en la Era del Big Data
Hadoop: Caso Práctico 1

Sara Del Río García

Departamento de Ciencias de la Computación e Inteligencia Artificial,
E.T.S. de Ingenierías Informática y de Telecomunicación,
srio@decsai.ugr.es