

## Big data: architectures and data analytics

## MapReduce - Exercises

2

### Exercise #5

- Average
  - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
    - Each line of the files has the following format  
sensorId,date,PM10 value ( $\mu\text{g}/\text{m}^3$ )\n
  - Output: report for each sensor the average value of PM10

3

### Exercise #5 - Example

- Input file
 

```
s1,2016-01-01,20.5
s2,2016-01-02,30.1
s1,2016-01-01,60.2
s2,2016-01-02,20.4
s1,2016-01-03,55.5
s2,2016-01-03,52.5
```
- Output pairs (s1, 45.4)  
(s2, 34.3)

4

### Exercise #6

- Max and Min
  - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
    - Each line of the files has the following format  
sensorId,date,PM10 value ( $\mu\text{g}/\text{m}^3$ )\n
  - Output: report for each sensor the maximum and the minimum value of PM10

5

### Exercise #6 - Example

- Input file
 

```
s1,2016-01-01,20.5
s2,2016-01-02,30.1
s1,2016-01-01,60.2
s2,2016-01-02,20.4
s1,2016-01-03,55.5
s2,2016-01-03,52.5
```
- Output pairs (s1, max=60.2\_min=20.5)  
(s2, max=52.5\_min=20.4)

6

## Exercise #7

- Inverted index
  - Input: a textual file containing a set of sentences
    - Each line of the file has the following format  
sentence*d*tsentence*n*
  - Output: report for each word *w* the list of sentence*ids* of the sentences containing *w*
    - Do not consider the words "and", "or", "not"

7

## Exercise #7 - Example

- Input file

```
Sentence#1 Hadoopor Spark
Sentence#2 Hadoopor Spark and Java
Sentence#3 Hadoopand Big Data
```

- Output pairs (hadoop, [Sentence#1, Sentence#2, Sentence#3])  
(spark, [Sentence#1, Sentence#2])  
(java, [Sentence#2])  
(big, [Sentence#3])  
(data, [Sentence#3])

8

## Exercise #8

- Total income for each month of the year and Average monthly income per year
  - Input: a (structured) textual csv files containing the daily income of a company
    - Each line of the files has the following format  
date*t*daily income*n*
  - Output:
    - Total income for each month of the year
    - Average monthly income for each year

9

## Exercise #8 - Example

- Input file

```
2015-11-01 1000
2015-11-02 1305
2015-12-01 500
2015-12-02 750
2016-01-01 345
2016-01-02 1145
2016-02-03 200
2016-02-04 500
```

- Output

```
(2015-11,2305) (2015, 1777.5)
(2015-12, 1250)
(2016-01, 1490) (2016,1095.0)
(2016-02, 700)
```

10