



Exercise #30

- Log filtering
 - Input: a simplified log of a web server (i.e., a textual file)
 - Each line of the file is associated with a URL request
 - Output: the lines containing the word "google"
 - Store the output in an HDFS folder

■ Input file 66.249.69.97--[24/Sep/2014:22:25:44+0000]"GET http://www.google.com/bot.html" 66.249.69.97-124/Sep/2014:22:56.4+0000]"GET http://www.google.com/bot.html" 66.249.69.97-124/Sep/2014:22:30:12+0000]"GET http://www.google.com/how.html" 71.19.157.179--[24/Sep/2014:22:30:12+0000]"GET http://www.google.com/faq.html" 66.249.69.97--[24/Sep/2014:23:28:44+0000]"GET http://dbdmg.polito.it/thesis.html" 66.249.69.97--[24/Sep/2014:22:25:44+0000]"GET http://www.google.com/bot.html" 66.249.69.97--[24/Sep/2014:22:25:44+0000]"GET http://www.google.com/how.html" 71.19.157.179--[24/Sep/2014:22:30:12+0000]"GET http://www.google.com/how.html"

Exercise #31

- Log analysis
 - Input: log of a web server (i.e., a textual file)
 - Each line of the file is associated with a URL request
 - Output: the list of distinct IP addresses associated with the connections to a google page (i.e., connections to URLs containing the term "www.google.com")
 - Store the output in an HDFS folder

Exercise #31 - Example

Input file

66.249.69.97 - [24/Sep/2014;22:25:44+0000] "GET http://www.google.com/bot.html"
66.249.69.97 - [24/Sep/2014;22:25:44+0000] "GET http://www.google.com/how.html"
66.249.69.97 - [24/Sep/2014;22:25:24+0000] "GET http://dbdmg.polito.it/course.html"
71.19.157.379 - [24/Sep/2014;22:20:21+0000] "GET http://dbdmg.polito.it/hesis.html"
66.249.69.95 - [24/Sep/2014;31:28:44+0000] "GET http://dbdmg.polito.it/hesis.html"

Output

66.249.69.97
71.19.157.379

Exercise #32

- Maximum value
 - 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 (μg/m³)\n
 - Output: report the maximum value of PM10
 - Print the result on the standard output

• Input file • Input file • S1,2016-01-01,20.5 \$2,2016-01-02,60.2 \$2,2016-01-02,00.2 \$1,2016-01-03,55.5 \$2,2016-01-03,55.5 \$2,2016-01-03,52.5

Exercise #33

- Top-k maximum values
 - 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 (μg/m³)\n
 - Output: report the top-3 maximum values of PM10
 - Print the result on the standard output

Input file \$1,2016-01-01,20.5 \$2,2016-01-01,30.1 \$1,2016-01-02,60.2 \$2,2016-01-02,20.4 \$1,2016-01-03,55.5 \$2,2016-01-03,52.5 Output 60.2 55.5 52.5

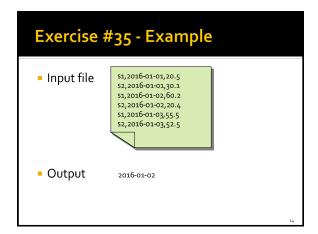
Exercise #34

- Readings associated with the maximum value
 - 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 (μg/m³)\n
 - Output: the line(s) associated with the maximum value of PM10
 - Store the result in an HDFS folder

Exercise #35

- Dates associated with the maximum value
- 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 (μg/m³)\n
- Output: the date(s) associated with the maximum value of PM10
 - Store the result in an HDFS folder

12



Exercise #36

- Average value
 - 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 (μg/m³)\n
 - Output: compute the average PM10 value
 - Print the result on the standard output