



- Word count problem
 - Input: (unstructured) textual file
 - Output: number of occurrences of each word appearing in the input file

Input file Toy example file for Hadoop. Hadoop running example. Output pairs (toy, 1) (example, 2) (file, 1) (for, 1) (hadoop, 2) (running, 1)

Exercise #2

- Word count problem
 - Input: a HDFS folder containing textual files
 - Output: number of occurrences of each word appearing in at least one file of the collection (i.e., files of the input directory)
- The only difference with respect to exercise
 #1 is given by the input
 - Now the input is a collection of textual files

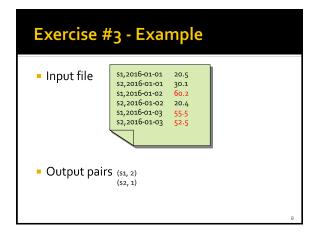
Exercise #2 - Example

Input files

Toy example file for Hadoop.
Hadoop running example.

Output pairs (another, 1) (example, 2) (file, 2) (for, 2) (hadoop, 3) (running, 1) (toy, 1)

- PM10 pollution analysis
 - Input: a (structured) textual file containing the daily value of PM10 for a set of sensors
 - Each line of the file has the following format sensorId,date\tPM10 value (μg/m³)\n
 - Output: report for each sensor the number of days with PM10 above a specific threshold
 - Suppose to set threshold = 50 μg/m³



Exercise #4

- PM10 pollution analysis per city zone
- Input: a (structured) textual file containing the daily value of PM10 for a set of city zones
 - Each line of the file has the following format zoneld,date\tPM10 value (µg/m³)\n
 - Output: report for each zone the list of dates associated with a PM10 value above a specific threshold
 - Suppose to set threshold = 50 μg/m³



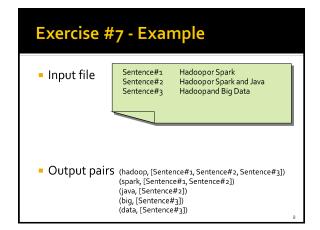


- Average
 - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
 - Each line of the files has the following format sensorId, date, PM10 value (μg/m³)\n
 - Output: report for each sensor the average value of PM10

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 (μg/m³)\n
 - Output: report for each sensor the maximum and the minimum value of PM10

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



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\tdaily income\n
 - Output:
 - Total income for each month of the year
 - Average monthly income for each year

Exercise #8 - Example 2015-11-01 Input file 2015-11-02 1305 2015-12-01 500 2015-12-02 2016-01-01 750 345 2016-01-02 1145 2016-02-03 200 2016-02-04 Output (2015-11,2305) (2015, 1777.5) (2015-12, 1250) (2016-01, 1490) (2016,1095.0) (2016-02, 700)





- Word count problem
 - Input: (unstructured) textual file
 - Output: number of occurrences of each word appearing in the input file
- Solve the problem by using in-mapper combiners

Input file Toy example file for Hadoop. Hadoop running example. Output pairs (toy, 1) (example, 2) (file, 1) (for, 1) (hadoop, 2) (running, 1)

Exercise #10

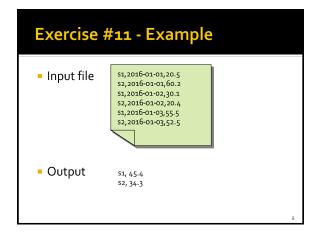
- Total count
 - Input: a collection of (structured) textual csv files containing the daily value of PM10 for a set of sensors
 - Each line of the files has the following format sensorId, date, PM10 value (μg/m³)\n
 - Output: total number of records

Exercise #10 - Example

Input file

| 51,2016-01-01,20.5 | 52,2016-01-02,30.1 | 52,2016-01-02,30.1 | 52,2016-01-02,30.5 | 52,2016-01-03,55.5 | 52,2016-01-03,52.5 |
| Output: 6

- 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 (μg/m³)\n
 - Output: report for each sensor the average value of PM10
 - Suppose the number of sensors is equal to 2 and their ids are s1 and s2



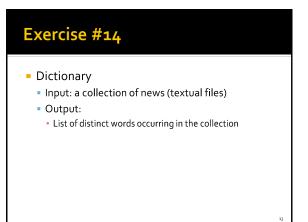
Exercise #12

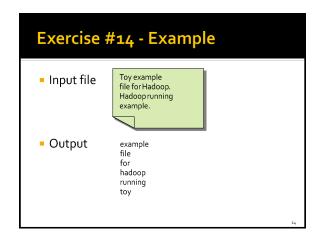
- Select outliers
 - Input: a collection of (structured) textual files containing the daily value of PM10 for a set of sensors
 - Each line of the files has the following format sensorId,date\tPM10 value (μg/m³)\n
 - Output: the records with a PM10 value below a user provided threshold (the threshold is an argument of the program)

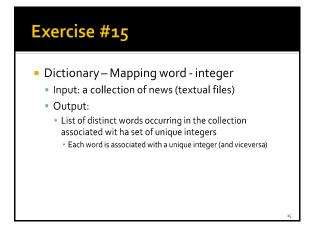
Exercise #13

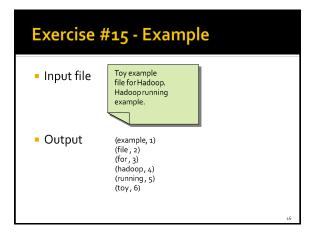
- Top 2 most profitable dates
 - Input: a (structured) textual csv files containing the daily income of a company
 - Each line of the files has the following format date\tdaily income\n
 - Output:
 - Select the date and income of the top 2 most profitable dates

Exercise #13 - Example 2015-11-01 1000 Input file 2015-11-02 1305 2015-12-01 500 2015-12-02 750 2016-01-01 2016-01-02 1145 2016-02-03 200 2016-02-04 500 Output 2015-11-02 1305 2016-01-02 1145













- Select maximum temperature for each date
 - Input: two structured textual files containing the temperatures gathered by a set of sensors
 - Each line of the first file has the following format sensorID, date, hour, temperature\n
 - Each line of the second file has the following format date, hour, temperature, sensorID\n
 - Output: the maximum temperature for each date (considering the data of both input files)

Input files
 \$1,2016-01-01,14:00,20.5\$
\$52,2016-01-01,14:00,30.2\$
\$51,2016-01-02,14:10,11.5\$
\$52,2016-01-02,14:10,30.2\$

 Output

2016-01-01,14:00,20.1,53
2016-01-02,14:15,31.5,53
2016-01-02,14:15,20.2,54
2016-01-02,14:15,20.2,54
2016-01-02,14:15,20.2,54
30.2
31.5

Exercise #18

- Filter the readings of a set of sensors based on the value of the measurement
 - Input: a set of textual files containing the temperatures gathered by a set of sensors
 - Each line of the files has the following format sensorID,date,hour,temperature\n
 - Output:
 - The lines of the input files associated with a temperature value greater than 30.0

- Filter the readings of a set of sensors based on the value of the measurement
 - Input: a set of textual files containing the temperatures gathered by a set of sensors
 - Each line of the files has the following format sensorID,date,hour,temperature\n
 - Output:
 - The lines of the input files associated with a temperature value less than or equal to 30.0

Exercise #20

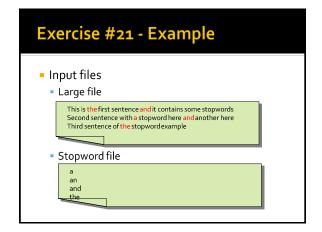
- Split the readings of a set of sensors based on the value of the measurement
 - Input: a set of textual files containing the temperatures gathered by a set of sensors
 - Each line of the files has the following format sensorID, date, hour, temperature\n
 - Output:
 - a set of files with the prefix "high-temp-" containing the lines of the input files with a temperature value greater than 30.0
 - a set of files with the prefix "normal-temp-" containing the lines of the input files with a temperature value less than or equal to 30.0

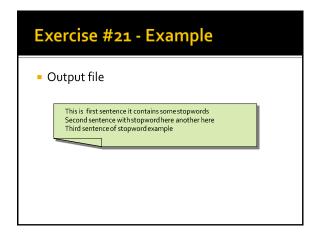
Input file s1,2016-01-01,14:00,20.5 s2,2016-01-01,14:00,30.2 s1,2016-01-02,14:10,31.5 s2,2016-01-02,14:10,30.2
 Output files high-temp-m-00001 normal-temp-m-00001 s2,2016-01-02,14:10,30.2 s2,2016-01-02,14:10,30.2





Stopword elimination problem Input: A large textual file containing one sentence per line A small file containing a set of stopwords One stopword per line Output: A textual file containing the same sentences of the large input file without the words appearing in the small file The order of the sentences in the output file can be different from the order of the sentences in the input file









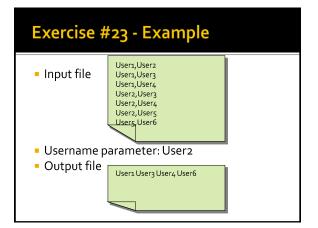
Friends of a specific user Input: A textual file containing pairs of users (one pair per line) Each line has the format Username1, Username2 Each pair represents the fact that Username1 is friend of Username2 (and vice versa) One username specified as parameter by means of the command line Output:

• The friends of the specified username stored in a textual file

· One single line with the list of friends

 Input file
 Usera, Userz Usera, Userz Usera, Users
 Username parameter: Userz
 Output file

Potential friends of a specific user
Input:
A textual file containing pairs of users (one pair per line)
Each line has the format
Usernamez, Usernamez
Each pair represents the fact that Username1 is friend of Username2 (and vice versa)
One username specified as parameter by means of the command line
Output:
The potential friends of the specified username stored in a textual file
Gine insigle line with the list of potential friends
Userz is a potential friend of User2 if they have at least one friend in common



Exercise #23 Bis

- Potential friends of a specific user
 - Solve problem #23 by removing the friends of the specified user from the list of its potential friends

Usera, Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera

Exercise #24

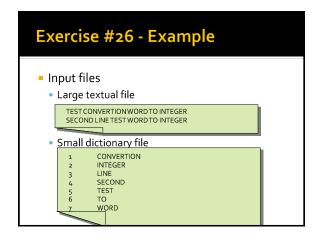
- Compute the list of friends for each user
 - Input:
 - A textual file containing pairs of users (one pair per line)
 - Each line has the format
 - Username1,Username2
 - Each pair represents the fact that Username1 is friend of Username2 (and vice versa)
 - Output:
 - A textual file containing one line for each user. Each line contains a user and the list of its friends

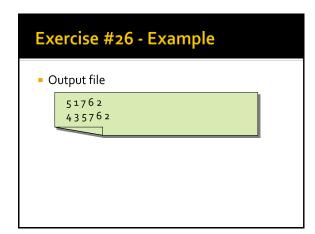
Input file
 Usera, Usera
 Usera, Usera
 Usera, Usera
 Usera, Usera
 Usera, Usera
 Usera: Usera
 Usera: Usera Usera
 Usera: Usera

Exercise #25

- Compute the list of potential friends for each
 - Input:
 - A textual file containing pairs of users (one pair per line)
 - Each line has the format
 - Username1, Username2
 - Each pair represents the fact that Username1 is friend of Username2 (and vice versa)
 - Output:
 - A textual file containing one line for each user with at least one potential friend. Each line contains a user and the list of its potential friends
 - Usera is a potential friend of Usera if they have at least one friend in common

Usera, Usera
Usera
Usera
Usera Usera Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera
Usera









- Categorization rules
 - Input:
 - A large textual file containing a set of records
 - Each line contains the information about one single user
 - Each line has the format
 - UserId,Name,Surname,Gender,YearOfBirth,City,Education
 - A small file with a set of business rules that are used to assign each user to a category
 - Each line contains a business rule with the format
 - Gender=<value> and Year Of Birth=<value> -> Category
 - Rules are mutually exclusive

Exercise #27

- Output:
 - One record for each user with the following format
 - The original information about the user plus the category assigned to the user by means of the business rules
 - Since the rules are mutually exclusive, there is only one rule applicable for each user
 - If no rules is applicable/satisfied by a user, assign the user to the "Unknown" category

4

Exercise #27 - Example

Users

User#1, John, Smith, M, 1934, New York, Bachelor User#2, Paul, Jones, M, 1956, Dallas, College User#3, Jenny, Smith, F, 1934, Philadelphia, Bachelor User#4, Laura, White, F, 1926, New York, Doctorate

Business rules

Gender=MandYearOfBirth=1934 -> Category#1 Gender=MandYearOfBirth=1956 -> Category#3 Gender=FandYearOfBirth=1934 -> Category#2 Gender=FandYearOfBirth=1956 -> Category#2

Exercise #27 - Example

Output

User#1, John, Smith, M, 1934, New York, Bachelor, Category#1 User#2, Paul, Jones, M, 1956, Dallas, College, Category#2 User#3, Jenny, Smith, F, 1934, Los Angleses, Bachelor, Category#2 User#4, Laura, White, F, 1926, New York, Doctorate, Unknown

- Mapping Question-Answer(s)

 - A large textual file containing a set of questions
 - Each line contains one question
 - Each line has the format
 - QuestionId,Timestamp,TextOfTheQuestion
 - A large textual file containing a set of answers
 - Each line contains one answer
 - Each line has the format
 - Answerld, QuestionId, Timestamp, TextOfThe Answer

Exercise #28

- Output:
- One line for each pair (question, answer) with the following format
 - QuestionId,TextOfTheQuestion, AnswerId,TextOfTheAnswer

Exercise #28 - Example Questions

Q1,2015-01-01,What is ..? Q2,2015-01-03,Who invented ..

Answers

A1,Q1,2015-01-02,It is .. A2, Q2, 2015-01-03, John Smith A3,Q1,2015-01-05,I think it is ..

Exercise #28 - Example

Output

Q1,What is ..?,A1,It is .. Q1,What is ..?,A3,I think it is .. Q2,Who invented ..,A2,John Smith

Exercise #29

- User selection
 - Input:
 - A large textual file containing a set of records
 - Each line contains the information about one single user
 - Each line has the format
 - UserId,Name,Surname,Gender,YearOfBirth,City,Education
 - A large textual file with pairs (Userid, MovieGenre)
 - Each line contains pair Userid, MovieGenre with the format
 - Userid, Movie Genre
 - · It means that UserId likes movies of genre MovieGenre

Exercise #29

- Output:
 - One record for each user that likes both Commedia and Adventure movies
- Each output record contains only Gender and YearOfBirth of a selected user
 - Gender, Year Of Birth
- Duplicate pairs must not be removed

