

Spark

Solutions Exercise 39 bis

Exercise #39 bis

- Critical dates analysis
 - Input: a textual csv file 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: an HDFS file containing one line for each sensor
 - Each line contains a sensorId and the list of dates with a PM₁₀ values greater than 50 for that sensor
 - Also the sensors which have never been associated with a PM₁₀ values greater than 50 must be included in the result (with an empty set)

Exercise #39 bis - Example

- Input file

```
s1,2016-01-01,20.5  
s2,2016-01-01,30.1  
s1,2016-01-02,60.2  
s2,2016-01-02,20.4  
s1,2016-01-03,55.5  
s2,2016-01-03,52.5  
s3,2016-01-03,12.5
```

- Output

```
(s1, [2016-01-02, 2016-01-03])  
(s2, [2016-01-03])  
(s3, [])
```

Exercise #39 bis – Solution V1

- From each input reading
 - if $PM_{10} > 50$ -> Return (sensorId, date)
 - if $PM_{10} \leq 50$ -> Return (sensorId, null)
- Group together values by key
 - Return (sensorId, [list of values])
 - One pair per distinct sensorId is returned
- Remove null from the lists of values

Exercise #39 bis – Solution V1

- N: number of input lines
 - S: number of distinct sensors
 - M: number of input lines with $PM_{10} > 50$
 - S_c : number of distinct sensors associated with $PM_{10} > 50$ at least on time
-
- $M < N$
 - $S_c < S$
 - $S \ll N$

Exercise #39 bis – Solution V1

textFile(..)

N strings



mapTopPair(...)

N pairs



groupByKey()

N pairs are sent on the network.


S pairs are created.



mapValues(...)

S values are updated

Exercise #39 bis – Solution V2

- Select only readings with $PM_{10} > 50$
 - From each of the selected input readings
 - Return (sensorId, date)
 - Group together values by key
 - Return (sensorId, [list of dates])
 - One pair per distinct sensorId is returned
- 
- From each input reading
 - Return sensorId
 - Remove duplicates
 - Apply subtract to remove the sensorIds that are associated with $PM_{10} > 50$ at least one time
 - For each of the selected sensorIds
 - Return (sensorIds, []), where [] is the empty list

└───────────┬───────────┘ Union ───────────┘

Exercise #39 bis – Solution V2

- N: number of input lines
 - S: number of distinct sensors
 - M: number of input lines with $PM_{10} > 50$
 - S_c : number of distinct sensors associated with $PM_{10} > 50$ at least on time
 - NPart: number of partitions
-
- $M < N$
 - $S_c < S$
 - $S \ll N$

Exercise #39 bis – Solution V2 – Part 1

textFile(..)



filter(..)



mapTopPair(...)



groupByKey()

N strings

$M < N$ strings are selected

M pairs

M pairs are sent on the network.

$S_c < S$ pairs are created.

Exercise #39 bis – Solution V2 – Part 2

textFile(..)

N strings

↓
map(..)

N strings

↓
distinct(..)

S x NPart values are sent
on the network.

S values are returned.

↓
subtract(... .keys())

Sc values are sent on the
network

↓
mapToPair()

S-Sc values are returned

Exercise #39 bis – Solution V2 – Part 3

`union(..)`

S pairs are returned

Comparison

- N: number of input lines
 - S: number of distinct sensors
 - M: number of input lines with $PM_{10} > 50$
 - S_c : number of distinct sensors associated at least on time with $PM_{10} > 50$
 - NPart: number of partitions
-
- $M < N$
 - $S_c < S$
 - $S \ll N$

Comparison

- Solution V_1 sends on the network
 - N pairs (sensorId, value) on the network
- Solution V_2 sends on the network
 - M pairs (sensorId, value) +
 - $S \times N_{Part}$ sensorIds +
 - S_c sensorIds

Comparison

- $S \ll N$
- If we suppose that $P_{M \geq 50}$ is a rare event
 - $M \ll N$
- Hence, Solution V2 sends less data on the network than Solution V1 if $M \ll N$
 - if $M \ll N \quad \Rightarrow \quad N > M + S \times N_{Part} + S_c$