

Spark Streaming

# Spark - Exercises

# Exercise #51

- Full station identification in real-time
- Input:
  - A textual file containing the list of stations of a bike sharing system
    - Each line of the file contains the information about one station
      - id\tlongitude\tlatitude\tname
  - A streaming of readings about the status of the stations
    - Each reading has the format
      - StationId,# free slots,#used slots,timestamp

# Exercise #51

- Output:
  - For each reading with a number of free slots equal to 0
    - print on the standard output timestamp and name of the station
  - Emit new results every 2 seconds

# Exercise #52

- High stock price variation identification in real-time
- Input:
  - A streaming of stock prices
    - Each input record has the format
      - Timestamp, StockID, Price

# Exercise #52

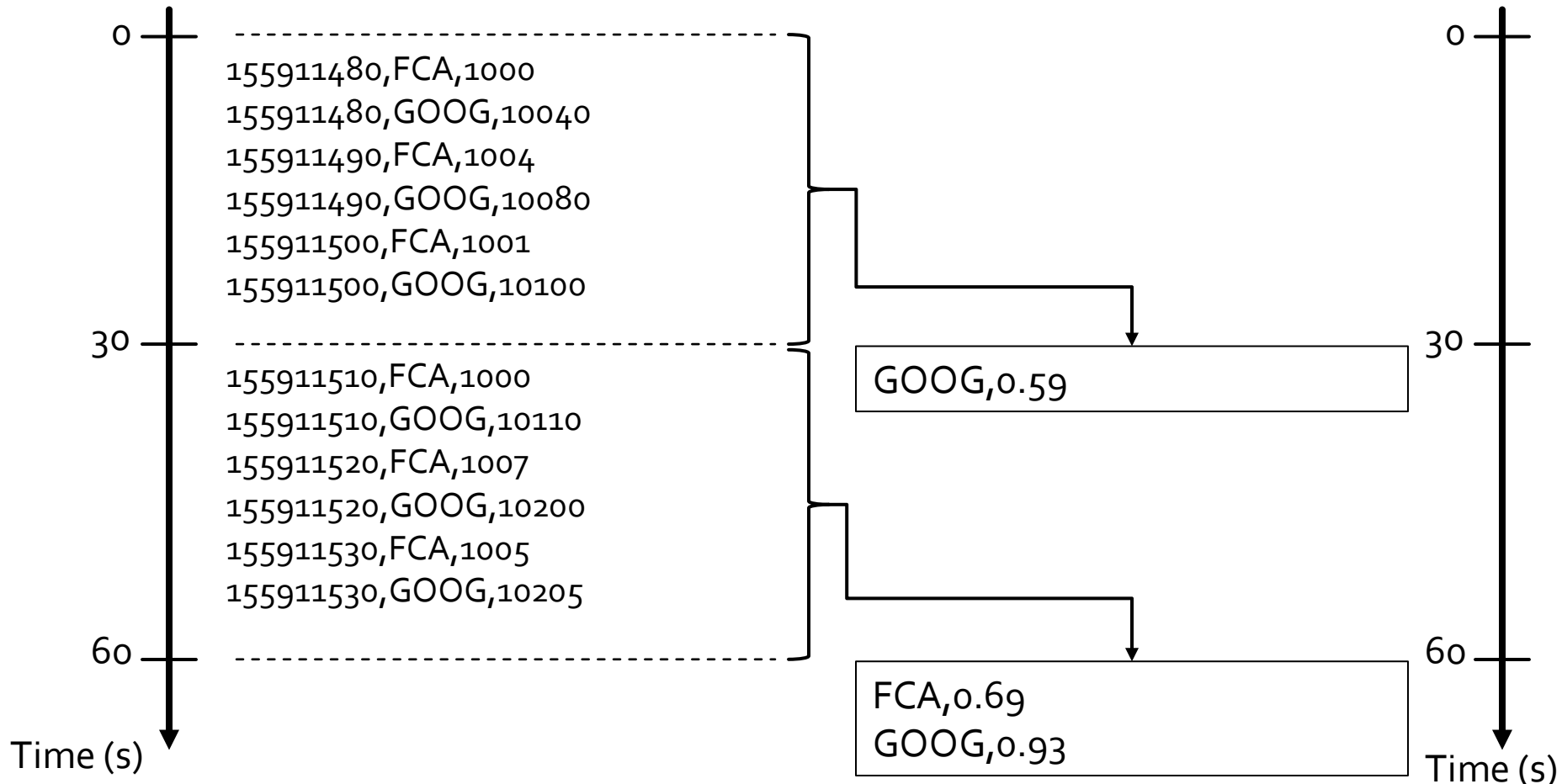
- Output:
  - Every 30 seconds print on the standard output, and store in the output folder, the **StockID and the price variation (%) in the last 30 seconds** of the stocks with a **price variation greater than 0.5%** in the last 30 seconds
  - Given a stock, its price variation during the last 30 seconds is:

$$\frac{\max(\text{price}) - \min(\text{price})}{\max(\text{price})}$$

# Exercise #52- Example

Input stream

Stdout



# Exercise #53

- Anomalous stock price identification in real-time
- Input:
  - A textual file containing the historical information about stock prices in the last year
    - Each input record has the format
      - Timestamp,StockID,Price
  - A real time streaming of stock prices
    - Each input record has the format
      - Timestamp,StockID,Price

# Exercise #53

- Output:
  - Every 1 minute print on the standard output, and store in the output folder, the StockIDs of the stocks that satisfy one of the following conditions
    - price of the stock (received on the real-time input data stream) < historical minimum price of that stock (based only on the historical file)
    - price of the stock (received on the real-time input data stream) > historical maximum price of that stock (based only on the historical file)
  - If a stock satisfies the conditions multiple times in the same batch, return the stockId only one time for each batch



# Exercise #53- Example

- Textual file containing the historical information about stock prices in the last year

130000000,FCA,1000

130000000,GOOG,10040

130000060,FCA,1004

130000060,GOOG,10080

130000120,FCA,1001

130000120,GOOG,10100

# Exercise #53- Example

Input stream

Stdout

