Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione

Esercitazione di gruppo Spark A.A. 2023/2024

Dataset

- **Download:** link
  - Versione (molto) ridotta, qualche MB
  - Reale circa 15GB
- **Entry point:** *'BDAchallenge2324'*
  - Una cartella per ogni anno
  - Per ogni anno, un file csv per ogni stazione (il nome del file è il nome della stazione)
    - Numero variabile di stazioni per ogni anno
    - NOTA: i file hanno alcuni campi non in comune, il programma deve gestire la situazione

Obiettivi

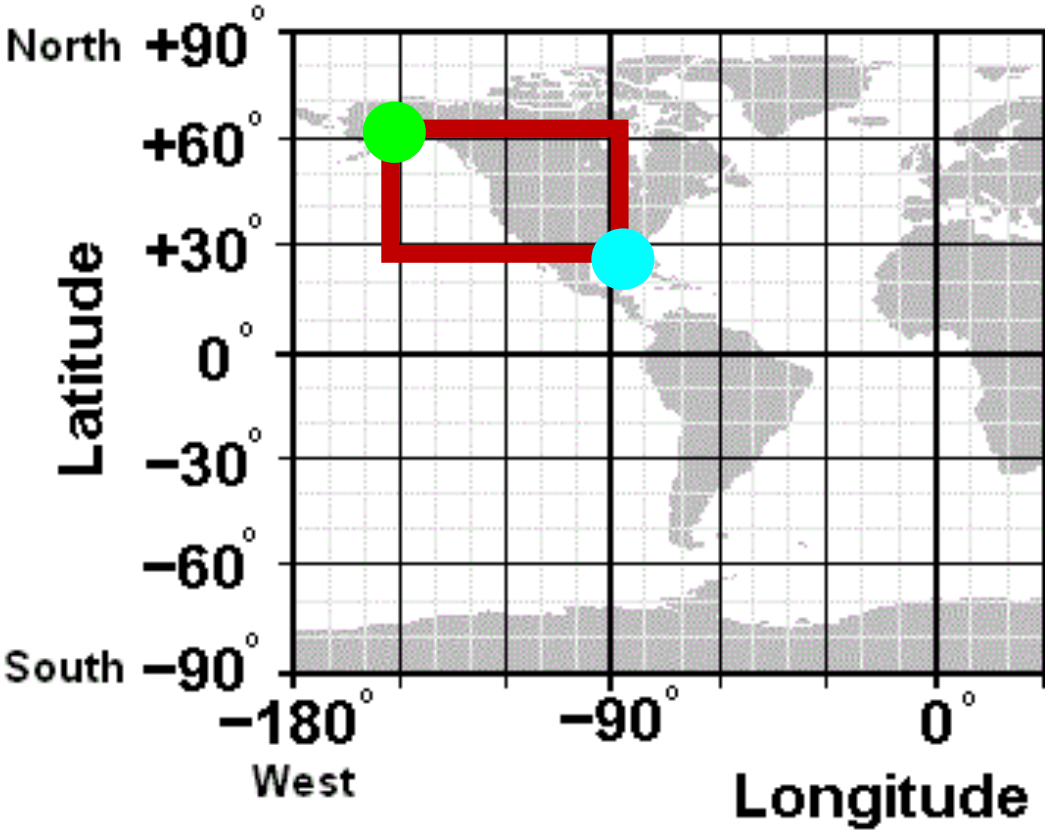
- 1 - Stampare il numero di misurazioni effettuate per ogni anno per ogni stazione (ordinato per anno e stazione)

```
2000,s_1,100
2000,s_2,98
```

Obiettivi

- 2 – Stampare le prime 10 temperature (TMP) con il maggior numero di occorrenze ed il relativo conteggio registrate nell'area evidenziata (ordinate per numero di occorrenze e temperatura)

```
[(60,-135);(30,-90)],22.1,40
[(60,-135);(30,-90)],21.17,30
```



Obiettivi

- 3 – La colonna `WND` contiene le informazioni relative al vento in formato «150,1,N,0041,1».
- L'elemento evidenziato rappresenta la velocità (in nodi) del vento. Stampare la stazione con la velocità in nodi che occorre più volte ed il relativo conteggio (ordinando per conteggio, velocità e stazione)

```
234234,9,40
```

Note

- **IMPORTANTE**: nella macchina per i test, i dati si trovano a partire da: *hdfs://192.168.104.45:9000/user/amircoli/BDA2324*
- **IMPORTANTE**: i file con i rusultati vanno memorizzati in: */home/amircoli/BDAchallenge2324/results/n* (n numero gruppo)

- **CONSEGNA**: unico file *n.ZIP* (n = numero gruppo) contenente
  - Presentazione della soluzione *(n.pdf/.ppt(x))*
  - File *n.py* con il codice della soluzione
  - File *.csv* dei risultati (un file per ogni richiesta, es: r1_n.csv)

Parti fisse codice


```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
entry_point = 'hdfs://192.168.104.45:9000/user/amircoli/BDAchallenge2324'

# your code
```

Suggerimento


```
# CONNECT WITH HDFS
fs = spark._jvm.org.apache.hadoop.fs.FileSystem.get(spark._jsc.hadoopConfiguration())

# GET FILES INFO STARTING FROM dir_path
list_status = fs.listStatus(spark._jvm.org.apache.hadoop.fs.Path(dir_path))

# GET FILENAMES
file_names = [files.getPath().getName() for files in list_status]
```

Punteggi

- **Efficacia**: 0.8 (0.4, 0.2, 0.2)
- **Efficienza**: 0.2

Buon lavoro!