Sistemi e Architetture per Big Data - A.A. 2019/2020
Progetto 2: Analisi dei ritardi e guasti del trasporto scolastico di NYC con Flink/Storm

Docenti: Valeria Cardellini, Fabiana Rossi
Dipartimento di Ingegneria Civile e Ingegneria Informatica
Università degli Studi di Roma "Tor Vergata"

Requisiti del progetto

Lo scopo del progetto è rispondere ad alcune query riguardanti un dataset relativo ai ritardi e guasti degli autobus che forniscono il servizio di trasporto scolastico per la città di New York, utilizzando il framework Apache Flink o, in alternativa, Apache Storm.

Il dataset contiene dati relativi a disservizi avvenuti dall'1 settembre 2015 al 27 novembre 2019 ed è composto da un unico flusso di input, fornito come file di testo in formato CSV e disponibile all'indirizzo `http://www.ce.uniroma2.it/courses/sabd1920/projects/prj2_dataset.zip`

Il flusso di input, contenuto nel file `bus-breakdown-and-delays.csv`, contiene diverse informazioni riguardanti l'occorrenza di ritardi e guasti ad autobus appartenenti a molteplici linee di trasporto scolastico che sono gestite da alcune compagnie; in particolare, ogni record rappresenta un evento di disservizio ed ha il seguente formato composto da 21 campi (tra parentesi è indicato il tipo di dato):

```
School_Year, Busbreakdown_ID, Run_Type, Bus_No, Route_Number, Reason,
Schools_Serviced, Occurred_On, Created_On, Boro, Bus_Company_Name,
How_Long_Delayed, Number_Of_Students_On_The_Bus,
Has_Contractor_Notified_Schools, Has_Contractor_Notified_Parents,
Have_You_Alerted_OPT, Informed_On, Incident_Number, Last_Updated_On,
Breakdown_or_Running_Late, School_Age_or_PreK
```

dove:

- `School_Year` indica l'anno scolastico (stringa di caratteri).
- `Busbreakdown_ID` è l'identificativo univoco dell'evento (intero).
- `Run_Type` indica se si è verificato un guasto o un ritardo su una categoria specifica di servizio di autobus (stringa di caratteri).
- `Bus_No` è il numero di autobus assegnato dal gestore della linea (stringa di caratteri).
- `Route_Number` è un identificatore univoco del numero del percorso (stringa di caratteri).
- `Reason` indica la causa del disservizio (stringa di caratteri). I motivi possibili sono: Accident, Delayed by School, Flat Tire, Heavy Traffic, Mechanical Problem, Other, Problem Run, Weather Condition, Won't Start.

- `Schools_Serviced` è una lista di codici delle scuole servite dall'autobus (stringa di caratteri).
- `Occurred_On` è il timestamp che indica quando il disservizio è avvenuto; il formato è YYYY-MM-DDThh:mm:ss.sss (stringa di caratteri).
- `Created_On` è il timestamp che indica quando il disservizio è stato registrato; il formato è YYYY-MM-DDThh:mm:ss.sss (stringa di caratteri).
- `Boro` indica il quartiere (o contea) in cui è avvenuto il disservizio (stringa di caratteri).
- `Bus_Company_Name` è il nome della compagnia di autobus (stringa di caratteri).
- `How_Long_Delayed` è il ritardo causato dal disservizio (stringa di caratteri).
- `Number_Of_Students_On_The_Bus` è il numero di studenti sull'autobus (intero).
- `Has_Contractor_Notified_Schools` indica se la compagnia ha avvisato la scuola riguardo il disservizio (stringa di caratteri).
- `Has_Contractor_Notified_Parents` indica se la compagnia ha avvisato i genitori degli alunni riguardo il disservizio (stringa di caratteri).
- `Have_You_Alerted_OPT` indica se la compagnia ha avvisato l'Office of Pupil Transportation (OPT) riguardo il disservizio (stringa di caratteri).
- `Informed_On` è la data in cui la scuola, i genitori oppure l'OPT sono stati informati; il formato è YYYY-MM-DDThh:mm:ss.sss (stringa di caratteri).
- `Incident_Number` è il numero identificativo dell'incidente eventualmente assegnato dal servizio clienti dell'OPT (stringa di caratteri).
- `Last_Updated_On` è la data in cui il record è stato aggiornato; il formato è YYYY-MM-DDThh:mm:ss.sss (stringa di caratteri).
- `School_Age_or_PreK` indica la tipologia di alunni servita dall'autobus (stringa di caratteri).

Gli eventi sono ordinati in base a quando sono accaduti (`Occurred_On`).

Il progetto è dimensionato per un gruppo composto da **2 studenti**; per gruppi composti da 1 oppure 3 studenti, si vedano le indicazioni specifiche. Supponendo di effettuare il replay del dataset (accelerando la scala temporale), si chiede di rispondere alla seguenti query in tempo reale:

1. Calcolare il ritardo medio degli autobus per quartiere nelle ultime 24 ore (di event time), 7 giorni (di event time) e 1 mese (di event time). L'output della query ha il seguente schema:

   ```
   ts, boro_x, avg_x, ..., boro_z, avg_z
   ```

   dove

   ```
   ts     // timestamp relativo all'inizio del periodo su cui e'
      ↪ calcolata la media
   boro_x // nome del quartiere x
   avg_x  // ritardo medio nel quartiere x
   ```

```
...
boro_z // nome del quartiere z
avg_z  // ritardo medio nel quartiere z
```

2. Fornire la classifica delle tre cause di disservizio più frequenti (ad esempio, Heavy Traffic, Mechanical Problem, Flat Tire) nelle due fasce orarie di servizio 5:00-11:59 e 12:00-19:00. Le tre cause sono ordinate dalla più frequente alla meno frequente. L'output della query ha il seguente schema:

   ```
   ts, slot_a, rank_a, slot_p, rank_p
   ```

   dove

   ```
   ts     // timestamp di inizio classifica
   slot_a // fascia oraria del mattino
   rank_a // classifica delle 3 cause nella fascia oraria del mattino
   slot_p // fascia oraria del pomeriggio
   rank_p // classifica delle 3 cause nella fascia oraria del
      ↪ pomeriggio
   ```

   La classifica dovrà essere calcolata sulle finestre temporali:

   - 24 ore (di event time),
   - 7 giorni (di event time).

3. Fornire la classifica in tempo reale delle 5 compagnie che hanno il punteggio di disservizio più alto. Il punteggio di disservizio viene calcolato come la somma pesata del numero di ritardi dovuti a "Heavy Traffic", "Mechanical Problem" e "Other Reason" (tutte le cause diverse da "Heavy Traffic" e "Mechanical Problem" devono essere classificate come "Other Reason", inclusa la causa Other) in base alla formula $w_t t + w_m m + w_o o$ dove t è il numero di ritardi dovuti a "Heavy Traffic", m è il numero di ritardi dovuti a "Mechanical Problem", o è il numero di ritardi dovuti a "Other Reason" e $w_t = 0.3, w_m = 0.5, w_o = 0.2$. Se la durata del ritardo è maggiore di 30 minuti, il corrispondente ritardo viene conteggiato due volte.

   L'output della classifica ha il seguente schema:

   ```
   ts, vendor_1, rating_1, vendor_2, rating_2, ..., vendor_5, rating_5
   ```

   dove

   ```
   ts       // timestamp di inizio classifica
   vendor_1 // id della compagnia classificata prima
   rating_1 // punteggio complessivo della compagnia classificata
      ↪ prima
   ...
   vendor_5 // id della compagnia classificata quinta
   rating_5 // punteggio complessivo della compagnia classificata
      ↪ quinta
   ```

   La classifica dovrà essere calcolata sulle finestre temporali:

- 24 ore (di event time),
- 7 giorni (di event time),

Gli output delle query devono anche essere memorizzati in file CSV e consegnati.

Si chiede inoltre di valutare sperimentalmente i tempi di latenza ed il throughput delle tre query durante il processamento sulla piattaforma di riferimento usata per la realizzazione del progetto, riportando tali tempi nella presentazione e nella relazione. La piattaforma di data stream processing può essere un nodo standalone con Apache Flink o Apache Storm oppure in alternativa è possibile utilizzare un servizio Cloud per stream processing (ad es. Amazon EMR con Flink o Google Dataflow), avvalendosi dei rispettivi grant a disposizione.

Opzionale: Rispondere ad una query a scelta tra le tre sopra descritte usando Kafka Streams oppure Spark Streaming e confrontare, sulla stessa piattaforma di riferimento, le prestazioni in termini di tempo di latenza e throughput delle query ottenute dai due framework.

Per gruppi composti da 1 studente: si richiede di rispondere alle query 1 e 2.

Per gruppi composti da 3 studenti: in aggiunta ai requisiti sopra elencati, si richiede di utilizzare Kafka Streams oppure Spark Streaming per rispondere alle tre query e di confrontare, sulla stessa piattaforma di riferimento, le prestazioni in termini di latenza e throughput con quelle ottenute dal primo framework scelto.

Svolgimento e consegna del progetto

Comunicare alle docenti la composizione del gruppo entro **lunedì 15 giugno 2020**.

Per ogni comunicazione via email è necessario specificare *[SABD]* nell'oggetto (subject) dell'email. Il progetto è valido **solo** per l'A.A. 2019/2020 ed il codice deve essere consegnato **entro venerdì 3 luglio 2020** per poter raggiungere il punteggio massimo.

La consegna del progetto consiste in:

1. link a spazio di Cloud storage o repository contenente il codice del progetto da comunicare via email **entro venerdì 3 luglio 2020**; inserire i risultati delle query in formato CSV in una cartella denominata `Results`.
2. relazione di lunghezza compresa tra le 3 e le 6 pagine, da inviare via email **entro lunedì 6 luglio 2020**; per la redazione si consiglia di usare il formato ACM proceedings (https://www.acm.org) oppure il formato IEEE proceedings (https://www.ieee.org);
3. slide della presentazione orale, da inviare via email alle docenti **dopo** lo svolgimento della presentazione.

La presentazione si terrà **martedì 7 luglio 2020**; ciascun gruppo avrà a disposizione **massimo 15 minuti**.

Valutazione del progetto

I principali criteri di valutazione del progetto saranno:

1. rispondenza ai requisiti;
2. originalità;

3. architettura del sistema e deployment;
4. organizzazione del codice;
5. efficienza;
6. organizzazione, chiarezza e rispetto dei tempi della presentazione orale.