Report del progetto di Big Data: analisi dei crimini nella città di Chicago

Alessia Ventani - Mat. 901809
Simone Venturi - Mat.

24 giugno 2020

Indice

1 Teachers' notes

The goal of the project is to assess the students' skills in writing jobs of low/medium complexity and to correctly reason about the jobs' performances. The projects must be agreed with the teachers (do not start without explicit consent) and it consists of:

- finding a complex-enough dataset: about 1GB, possibly consisting of more tables;
- loading the dataset on HDFS/Hive;
- implement an analytical job in both MapReduce and Spark;
- writing a short report to describe it.

To deliver the project, each group must **send by email to both teacher the link to their repository**. The repository must:

- be created from the individual assignment available on the Github classroom;
- contain a report folder with the PDF of the final report (based on this template). The report can be written in either Italian/English and Latex/Word at your discretion. Be concise and go straight to the point: an excessively verbose report is a waste of time for you and for the teachers;
- contain a README.md file with the instruction to run the jobs. Indeed, the teachers must be able to clone the repository and run the jobs from their accounts on the cluster. This means that the dataset must be accessible on HDFS/Hive and the code must compile and run correctly. Please make the jobs repeatable (i.e., the job checks and possibly deletes old data to avoid errors when re-running the code).

This guide is based on the "MapReduce+Spark" kind of project. However, we remind that a different kind of project may be agreed upon.

The evaluation will be based on the following.

- Compliance of the jobs with the agreed upon specifications.
- Compliance of the report with this guide.
- Job correctness.
- Correct reasoning about optimizations.

Appreciated aspects.

- Code cleanliness and comments.
- Further considerations in terms of job scalability and extensibility.

2 Introduzione

2.1 Descrizione del dataset

Please provide:

- A brief description of the dataset.
- The link to the website publishing the dataset (e.g., `https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page`).
- Direct links to the downloaded files, especially if more than one files are available in the previous link (e.g., `https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2017-01.csv`).

Per questo elaborato si è deciso di utilizzare gli open data che la città di Chicago rende accessibili on-line. Fra i vari dataset presenti, ci si è concentrati su quello riportante i crimini commessi nella città Chicago dal 2001 ad oggi, aggiornati a sette giorni precedenti la data del download. I dati sono estratti dal CLEAR, acronimo di Citizen Law Enforcement Analysis and Reporting, del dipartimento di polizia della città. Per motivi ovvi di privacy, i nomi propri sono omessi e gli indirizzi non riconducono ad una specifica posizione geografica ma ad un'area, più o meno grande in base al grado di granularità del campo, della città.
I dati utilizzati sono scaricabili ai seguenti link:

- elenco dei crimini registrati: `https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2`
- elenco dei codici univoci di report dei crimini dello stato dell'Illinois: `https://data.cityofchicago.org/Public-Safety/Chicago-Police-Department-Illinois-Uniform-c7ck-438e`

Dai link riportati è possibile scaricare i dati premendo sul tasto Export e scegliendo il formato "CSV Excel for Europe".

2.1.1 Descrizione dei file

For each file, briefly indicated the available data and the fields used for the analyses; examples are welcome.

L'analisi si basa sull'utilizzo di un unico file csv di partenza formato con i dati dei crimini registrati a Chicago. Nella tabella sono presenti 22 colonne che riportano informazioni di varie categorie: le informazioni riguardanti il tipo di codice, la sua collocazione temporale e spaziale (comprese le coordinate), se è stato effettuato un arresto o meno e se il crimine è domestico. Di queste colonne sono stati considerati solo alcuni campi.
Per il primo job sono stati utilizzati:

- IUCR: codice univoco di identificazione di un crimine per lo stato dell'Illinois;

- Description: breve descrizione del crimine riportato;
- District: codice corrispondente al distretto di polizia in cui è avvenuto il crimine.

Per la seconda elaborazione ci si è concentrati su:

- IUCR: codice univoco di identificazione di un crimine per lo stato dell'Illinois;
- Description: breve descrizione del crimine riportato;
- Arrest: booleano che indica se per il crimine è stato effettuato un arresto o meno;
- Year: anno in cui è avvenuto il crimine.

Un esempio di dati riportati è:
IUCR Description District Arrest Year
0460 SIMPLE 006 False 2020

3 Preparazione dei dati

Please provide:

- The paths to each file on HDFS and/or its corresponding location in Hive (database and table); consider relying on the structured data lake organization.
- A subsection with details on the pre-processing of the data (only necessary if the data is dirty and/or it contains a significant amount of useless information).

I dati considerati non hanno avuto bisogno di operazioni di pre-elaborazione complesse data la loro natura. Si noti però che all'interno del dataset, essendo compilato manualmente, potrebbero esserci degli errori accidentali nei dati ma questo non costituisce un grosso problema per le elaborazioni che si intendono eseguire.

Per poter effettuare dei confronti nel primo job, vedi descrizione nella sezione successiva, sono state preparate due versioni del file inziale. La prima costituisce il file scaricato dal sito nella sua versione integrale mentre per la seconda si è deciso di dividere la tabella iniziale in due:

- la prima comprende tutti i campi ad eccezione di "Primary Type" e "Description". Per ottenere questo file è stato fatto eseguire il seguente codice spark:

```
spark.read.format("csv")
  .option("sep", ";").option("header", "true")
  .option("mode", "DROPMALFORMED")
  .load(<path to input file>)
  .drop("Primary Type","Description").coalesce(1)
  .write.format("com.databricks.spark.csv")
  .option("sep", ";").option("header", "true")
  .save(<path to output file>)
```

- la seconda riporta i campi "IUCR","Primary Type" e "Description". In questo caso non è stato necessario elaborare il file di partenza ,a è stato semplicemente scaricato l'elenco dei codici univoci di report dei crimini dello stato dell'Illinois. In quest'ultimo però alcuni codice IUCR non presentavano lo zero iniziare e quindi per farli coincidere con quelli presenti nella prima si è dovuto aggiungere questa cifra, operazione effettuata manualmente dato il basso numero di record con questa caratteristica.

Come formato si è deciso di caricare il file csv e utilizzare come separatore il simbolo ; per facilitare le operazioni nel paradigma map reduce.

I file sono stati caricati in hdfs e sono raggiungibili a:

4 Jobs

One subsection for each job.

4.1 Job #1: short description

Provide a brief, general description of the job. Then, one subsubsection for each implementation.

4.1.1 MapReduce/Spark implementation

Please provide:

- Input and output files/tables.
- Execution time and amount of resources.
- Direct links to the application's history on YARN (e.g., `http://isi-vclust0.csr.unibo.it:18088/history/application_15...`).
- Description of the implementation. A schematic and concise discussion is preferrable to a verbose narrative. Focus on how the data is manipulated in the job (e.g., what do keys and values represent across the different stages, what operations are carried out).

- Performance considerations with respect the (potentially) carried out optimizations, e.g., in terms of:
  - allocated resources and tasks;
  - enforced partitioning;
  - data caching;
  - combiner usage;
  - broadcast variables usage;
  - any other kind of optimization.
- Short extract of the output and discussion (i.e., whether there is any relevant insight obtained).

5 Miscellaneous

If necessary, feel free to add sections to explain any other relevant information.