TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Macroarea di Ingegneria
Dipartimento di Ingegneria Civile e Ingegneria Informatica

Project 1

Corso di Sistemi e Architetture per Big Data

A.A. 2019/2020

Valeria Cardellini, Fabiana Rossi

Laurea Magistrale in Ingegneria Informatica

Project delivery

- Submission deadline
  - May 29, 2020
  - After the deadline, the maximum achievable score will be decreased by 2 points for each week of delay
- Your presentation
  - June 4, 2020
- What to deliver
  - Link to cloud storage or repository containing project code
  - Project report composed by 3-6 pages in ACM or IEEE proceedings format (only the report: by June 1)
  - Presentation slides (max. **15 minutes** per group), to be delivered after your presentation
- Team
  - Target: 2 students per team
  - Also possible 1 student or 3 students per team

Dataset

- You will use two real datasets on Covid-19
- Both datasets are available in CSV format and are updated each day
- Datasets are available at:
  - Dataset 1 (Italy) provided by the Italian Civil Protection Department

    https://github.com/pcm-dpc/COVID-19/blob/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv
  - Dataset 2 (world) provided by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, USA

    https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv

Dataset 1: schema

- Input in CSV format
- One row per day, starting from February 24, 2020
- On each row:
  - Date with format *yyyy-mm-dd*T*hh:mm:ss*
  - Country (ITA)
  - Patients in hospitals with symptoms
  - Patients in intensive care
  - Total number of hospitalized patients
  - Home isolation
  - Total number of current confirmed cases
  - Change of total confirmed cases on a daily basis
  - Daily new confirmed cases
  - Number of recovered cases
  - Total number of deaths
  - Total cases (including recoveries and deaths)
  - Number of swab tests

Most data are cumulative: next day's values are the previous day's plus the new cases

Dataset 2: schema

- Input in CSV format
- One row per day, starting from January 22, 2020
- On each row:
  - Province/State
  - Country
  - Latitude and longitude
  - Daily columns with total number of confirmed cases

Queries with Hadoop/Spark

- Use the Hadoop framework (and the MapReduce programming model) or alternatively the Spark framework to answer some queries on the dataset
- Include in your report/slides the queries' response time on your reference architecture

Query 1

Using dataset 1, for each week determine the average number of cured people and the average number of swab tests

Queries with Hadoop/Spark

Query 2

Using dataset 2, for each continent determine the average, standard deviation, minimum and maximum number of confirmed cases on a weekly basis

- Consider only the top-100 affected states (if State field is not indicated, use Country)
- First use trendline coefficient to identify the top-100 affected states and then calculate statistics on a weekly basis
- Continent must be also identified
  - Consider 6 continents: Africa, America, Antarctica, Asia, Europe, Oceania

Queries with Hadoop/Spark

Query 3

Using dataset 2 and the top-50 affected states identified using the trendline coefficient, for each month use K-means clustering algorithm (with K=4) to identify the states that belong to each cluster with respect to the trend of confirmed cases

- Consider only the top-50 affected states (if State field is not indicated, use Country) every month; identify them using the trendline coefficient on the number of confirmed cases
- Run the clustering algorithm on each month to identify those states having a similar trend of confirmed cases during that month
- Compare the performance of a naïve implementation of K-means with that provided by Spark MLlib or Apache Mahout

Optional part A

- Compulsory for team composed of 3 students
- Use a higher level framework (Hive, Pig or Spark SQL) to address Queries 1 and 2
- Include in the report the query times using a higher level framework on your reference architecture and compare them to those achieved by your pure Hadoop/Spark-based solution

Data acquisition and ingestion

- Which framework to ingest data into HDFS?
  - Flume, NiFi, Kafka, …
- Which format to store data?
  - csv, columnar format (Parquet), row format (Avro), …
- Where to export your results?
  - HBase, Redis, Kafka, …

Optional part B

- Use a visualization software (e.g., Grafana) to graphically present the query results

Team composition and tasks

- 1 student in the team:
  - Queries 1 and 2
  - Data acquisition is optional
- 2 students in the team:
  - Queries 1, 2 and 3
  - Plus data acquisition
- 3 students in the team:
  - Queries 1, 2 and 3
  - Plus data acquisition
  - Plus optional part A using a higher level processing framework