Università degli Studi di Roma “Tor Vergata”
Dipartimento di Ingegneria Civile e Ingegneria Informatica

Project 1

Corso di Sistemi e Architetture per Big Data

A.A. 2017/18

Valeria Cardellini, Matteo Nardelli

Project delivery

- Submission deadline
  - June 1, 2018
  - After the deadline, the maximum achievable score will be decreased by 2 points for each week of delay
- Your presentation
  - June 7, 2018
- What to deliver
  - Link to cloud storage or repository containing the project code
  - *Optional*: project report composed by 4-6 pages in ACM or IEEE proceedings format
  - Slides of your presentation (max. **15 minutes** per group), to be delivered after the presentation
- Team
  - Target: 2 students per team
  - Also possible 1 student or 3 students per team

Dataset

- You will use a real dataset from ACM DEBS 2014 Grand Challenge
- Smart homes
  - Goal: batch analytics of energy consumption measurements over high volume sensor data
  - Reduced data set available at http://www.ce.uniroma2.it/courses/sabd1718/resources/debs14_reduced.tar.gz

Dataset

- Recordings originating from smart plugs
- Smart plug: a proxy between the wall power outlet and the device connected to it
  - Equipped with a range of sensors which measure different, power consumption related, values
- Smart plugs deployed in private households with data being collected roughly every 20 s for each sensor in each smart plug
  - Uncontrolled, real-world environment: possibility of malformed data as well as missing measurements

Dataset

- Hierarchical structure within a house
  - Identified by a unique house id
  - Every house contains one or more households, identified by a unique household id (within a house)
  - Every household contains one or more smart plugs, each identified by a unique plug id (within a household)
- Every smart plug contains two sensors
  1. load sensor measuring current load in Watt
  2. work sensor measuring total accumulated work since the start (or reset) of the sensor in kWh

Dataset: schema

- Input in csv format
- Each row contains:

  id, timestamp, value, property, plug_id, household_id, house_id
  - *id*: a unique identifier of the measurement [32 bit unsigned int]
  - *timestamp*: timestamp of measurement (number of seconds since January 1, 1970, 00:00:00 GMT) [32 bit unsigned int]
  - *value*: the measurement [32 bit floating point]
  - *property*: type of the measurement: 0 for work or 1 for load [boolean]
  - *plug_id*: a unique identifier (within a household) of the smart plug [32 bit unsigned int]
  - *household_id*: a unique identifier of a household (within a house) where the plug is located [32 bit unsigned int]
  - *house_id*: a unique identifier of a house where the household with the plug is located [32 bit unsigned int]

Dataset: schema

- Example of the dataset

Listing 2: Snapshot of the base stream data

```
1 2967740693,1379879533,82.042,0,1,0,12
2 2967740694,1379879533,105.303,1,1,0,12
3 2967740695,1379879533,208.785,0,2,0,12
4 2967740696,1379879533,217.717,1,2,0,12
5 2967740697,1379879533,2.207,0,3,0,12
6 2967740698,1379879533,2.081,1,3,0,12
7 2967740699,1379879533,0,1,3,1,12
8 2967740700,1379879533,0.313,0,3,1,12
9 2967740701,1379879533,0,1,3,2,12
```


Queries with Hadoop/Spark

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

1. Identify the houses with instant load greater than or equal to 350 Watts
2. For each house, calculate the average energy consumption and its standard deviation in the following four time slots: night (from 00:00 to 05:59), morning (from 06:00 to 11:59), afternoon (from 12:00 to 17:59), and evening (from 18:00 to 23:59)

Queries with Hadoop/Spark

3. Considering peak hours (Monday to Friday from 06:00 to 17:59) and off-peak hours (night time from Monday to Friday from 18:00 to 05:59, on weekends (Saturday and Sunday) and holidays), calculate the ranking of smart plugs based on the difference in the average monthly energy consumption between the peak hours and the off-peak hours
   - In the ranking the smart plugs are ordered in descending order, reporting, as first elements, the plugs that do not take advantage of the off-peak hours

Optional part

- Compulsory for team composed of 3 students
- Use either Hive (or Pig) or Spark SQL to address the same three queries
- 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

Queries for the team

- 1 student in the team: queries 1 and 3
- 2 students in the team: all the three queries
- 3 students in the team: all the three queries plus optional part

Data ingestion

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