Big Data - Part Data Management

Project Assignment - Phase I

2018-2019

Assignment

The *Internet of Things* (IoT for short) refers to the growing tendency to embed computing devices in everyday objects (cars, fridges, thermostats, ...) and to enable these devices to send and receive data through connection with the Internet.

One use case in the Internet of Things is the creation of *smart homes* and *smart energy monitoring*. The idea there is to equip households with various kind of sensors (temperature, humidity, presence, heating system consumption, ...) whose readings are communicated to order to allow fine-grained energy monitoring and — eventually — forecasting energy needs.

The Brussels government, wanting to lay claim to the title of *smart city*, has initiated a project where it will equip a modest number of households and public buildings with such sensors (exact description of the sensors in Section 1). In tandem, your team has been contracted to design and implement a Big Data Management & Analytics (BDMA) platform that will store, manage and analyze this sensor data.

The project consists of two phases:

1. In Phase I, you need to design the Big Data Management pipeline that will be used to store and manage the sensor data, as well as implement certain batch and streaming queries for the pipeline in order to show its feasibility.
2. In Phase II, you will employ Machine Learning methods on actual sensor data in order to build a model of the data that can be used for predictions.

The two phases have different objectives and different deadlines. In particular, this document only contains the tasks related to phase I. The assignment related to phase II will be published later in the semester.

Follow the outline of tasks below. You need to document your findings in a report that needs to be handed in together with your implementation.

1 The Data

During the project, different kinds of *sensors* will be placed in different kind of *spaces*.

A space is meant to represent a physical space (such as an apartment, a house, a floor in a public building, etc). Each space is given an internal id. Meta-data about each space such as, e.g., the address, the Brussels municipality[1] that hosts the space, whether the space is private (a home) or public (e.g., a work space), etc. must be recorded.

Each sensor has a unique identifier, and is either:

[1] `https://en.wikipedia.org/wiki/List_of_municipalities_of_the_Brussels-Capital_Region`

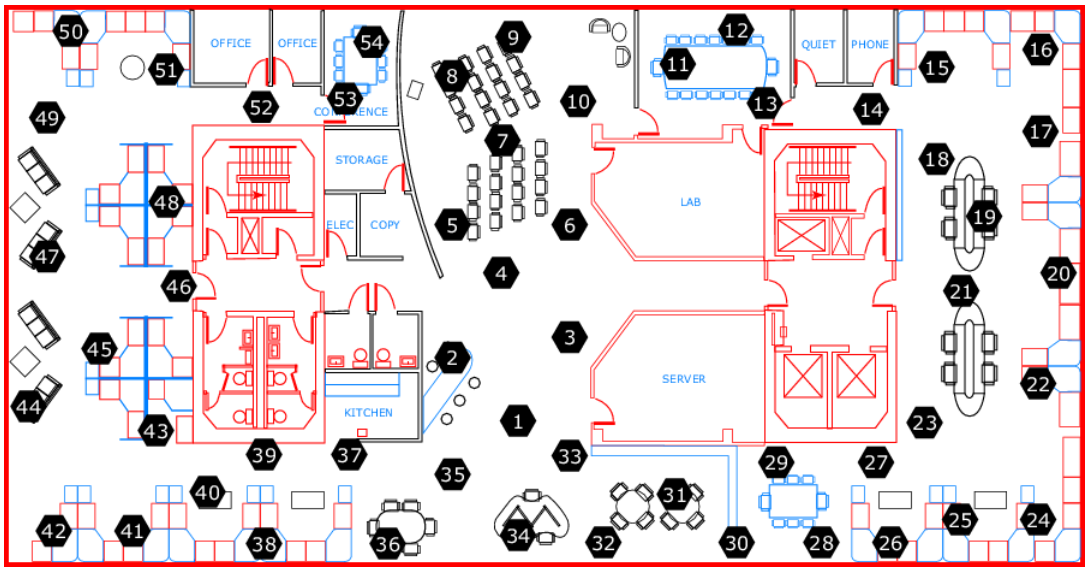

Figure 1: Placement of sensor points in the example dataset space.

– A temperature sensor, which sends information concerning the space's temperature (in $°C$, floating point).
– A humidity sensor, which measures relative humidity (in %, floating point)
– A light sensor, which measures luminosity (in Lux, floating point)
– A movement sensor, which measures if movement has been detected since the last time a measurement was reported (a Boolean)

In addition to these readings, a sensor will also report on the Voltage of its battery (in V, floating point).

For each participating space, the identifiers of the sensors located in the space must be stored, as well as their coordinates (x and y, w.r.t. some "base coordinate" that is stored for each space). For each sensor, it type (temperature, humidity, light, ...) also needs to be stored.

Sensors are expected to send a measurements once every $\pm 30s$; initially, 10 000 spaces are participating to the project, each equipped with all 4 types of sensors.

To illustrate the kind of data that you can expect, the course webpage contains an example log of sensor readings collected in a single public space (a research lab). The research lab was divided into 54 points as illustrated in Figure 1. Each point p was equipped with 3 sensors:
- Sensor with id p-0 measured temperature
- Sensor with id p-1 measured humidity
- Sensor with id p-2 measured light

Note that no movement sensors were deployed in this dataset. The x and y coordinates of points (in meters relative to the upper right corner of the research lab) are also available on the course webpage.

The log file is a text file with the following schema:
- date: yyyy-mm-dd
- time: hh:mm:ss.xxx
- sensorid: p-i with p an integer in the range [1-54] and i in the range $[0-2]$

- measurement:real, the sensor measurement
- voltage:real, the sensor voltage value at the time of measurement (in volt)

Note that some measurements are missing in the log file due to sensor failings. (I.e., not every sensor reports a value every $\pm 30s$).

2 Phase 1 Requirements

The objective in Phase I of the project is to design BDMA pipeline and implement an associated web-based dashboard in which the following information can be inspected:

1. Basic statistics (min, max, avg) about sensor readings (per type), grouped according to different granularities in space (per space, per municipality, or for the entirety of brussels) and time (for the last 24 hours, the last 2 days, the last week, the last month, the last year).

2. For each temperature sensor, a chararcterization of the day's timeslots, at a 15-minute granularity, in *daytime temperature* (temperature readings of at least $19.5°C$ have been observed) and *nighttime temperature* ($< 19.5°C$). This characterization should also allow grouping according to space (i.e. per space/municipality/Brussels as a whole) as well as type of space (public/private) and time (last month/last year).

   Concretely, there are 24h in a day. Divide these 24h into slots of 15 minutes. So, slot 1 is from 00h00 to 00h15; slot 2 from 00h15 to 00h30; and so on. You are hence asked to compute, for each day d, each slot s, and each temperature sensor x, the average temperature measured by x during slot s on day d, and use this to classify the slot s of day d into day or night.)

   The grouping requirement asks that you devise a method by which the results of (2) can be "rolled-up" to different time granularities (e.g. a classification of slot s based on the date of the entire last week/month/year) and different space granularities (e.g., a classification of the slot based on data not only from sensor s, but all sensors in a specific space/all sensors in a specific municipality/ ...

3. For each sliding window of 1h over the stream of all the received temperature measurements, an overview of the measurements that are frequent within the 1h sliding window, together with an estimate of their frequency. You are allowed to choose the frequency treshold (i.e., you are free to determine when a measurement is considered frequent). Your solution (as well as the frequencies reported) are allowed to be approximate. Temperature measurements can be rounded to a precision of one fractional digit. (E.g, 19.439 and 19.442 can be considered the same measurement).

Please take into account the following considerations in your design.

– New data is arriving in a streaming fashion (one measurement every $\pm 30s$ for every sensor).
– To be compatible with phase II of the project, all queries/analysis that you do on streaming data should be implemented using Spark Streaming.
– Old data should not be discarded. In the future one may want to go as far back as e.g., 10 years.
– Your dashboard needs to visualize new data as fast as possible. (I.e., it cannot work exclusively in batch mode).

– You need to describe in your report which sets of technologies you will use in your BDMA (how will the data be stored? what framework will be used to compute the answers to the queries above? How will these answers be stored? How will new data be made available to the sytem ? ...)
– We have described the λ-architecture in Lecture 3. Other Big Data architectures have been proposed as a response to the λ-architecture. As part of the project, you should research these other possible architectures, and evaluate which architecture you will use (+ give corresponding motivation).
– Your report needs to specify any additional assumption that you make w.r.t. the data.
– Your design needs to accommodate for future growth (i.e., be ready to scale to all households in Brussels). In this respect, the city of Brussels has explictly requested that your report include a section that analyzes the expected data volumes (and associated storage requirements) of your BDMA in function of the number of spaces and number of sensors per space.

3 Deliverables

For phase I, you need to deliver both your implementation of the BDMA pipeline (consisting of the code that executes the queries above as well as the code that runs the dashboard) and a report that documents your analysis, motivate the design that you have opted for, list and motivate the set of technologies that you have chosen, and describes your implementation.

Phase I will be evaluated based solely on your implementation and your report. This part of the project does not have an oral defense. In contrast, Phase II will have an oral defense, which will be scheduled in the exam session.

The code needs to be accompanied with a README file that explains how to set up the software required for running your code.

The report needs to discuss, at a minimum, the following items:

- The overall architecture of your envisioned BDMA pipeline (e.g., an instantiation of the λ-architecture, or some other kind of architecture).
- For each component in this architecture:
  - a description of its functional purpose,
  - a discussion of the set of possible technologies that were considered to implement the function
  - the technology that you selected, with a motivation of why it was selected over the alternatives.
- Additional assumptions that you make w.r.t. the data.
- The requested analysis of the data volumes (and associated storage requirements) of your BDMA.
- For each of the queries listed above, a short description of how they were implemented.
- A description of how the dashboard was implemented and screenshot of the dashboard.

Careful! Do not simply copy-and-paste text from the internet to include in your report. If you copy-and-paste existing text/graphics without duly citing the source from which this was copied this constitutes fraud, which will result on 0/20 on the course and possibly an exclusion from other exams!

Implementation environment. You must implement your BDMA by installing all required software in a locally-hosted environment. You may wish to use Docker (`https://www.docker.com/`) or Vagrant (`https://www.vagrantup.com/`) to facilitate sharing of the development environment

4 Modalities

The assignment has the following modalities:

1. The project assignment contributes 15/20 to the overall grade: 5/20 for phase 1, and 10/20 for phase 2. The written exam contributes the remaining 5/20 points.

2. Phase 1 of the project will be graded on (1) the implementation itself and (2) the report that you need to write to describe your analysis and motivate your design and implementation.

3. The project should be solved in groups of 2 (if the total number of students in the course is not divisible by 2, at most one groups of 3 students will be allowed). You are asked to register, per group, the names of the group members via the online poll available at

   `https://docs.google.com/forms/d/e/1FAIpQLSdHWvuVz2r3dxMG1DCsrKg7NOyT34cn-2f17KCBM7ur9WGcvw/viewform`

   by March 18 at the latest. If you cannot find a partner, please indicate so by sending an email to prof. Vansummeren (`svsummer@ulb.ac.be`), who will hook you up with a partner.

4. This project is mandatory. If you do not make the project, you cannot pass the course.

5. You will have to create a GIT repository[2] in the `INFO-H-515/2018-2019-phase-1-s1` repository group at `http://wit-projects.ulb.ac.be/rhodecode/` to submit both your report and your code. The username and password to login to this system correspond to your ULB/VUB NetID. The repository will be named

   `project-<student1>-<student2>`

   where `student` corresponds to your student number and `<student1>-<student2>` appear in sorted order. This repository must be made private. It is recommended that you create this repository *as soon as possible* to avoid last minute technical difficulties, and that you use it throughout the project to synchronize your changes.

   Tip. If you attempt to push a large set of changes to a GIT repository with HTTP or HTTPS, you may get an error message such as `error: RPC failed; result=22, HTTP code = 411.` This is caused by a GIT configuration default which limits certain

[2] `http://git-scm.com/documentation`

Art.34 In case of fraud or plagiarism during an examination or during a test at an interim date during the academic year, or in relation with the preparation of written reports or papers, the course professor reports the case in writing prior to the jury deliberation to the relevant academic authority levels for disciplinary matters. A copy of that fraud report is addressed to the jury chairmen. The student can ask to be heard by a jury chairperson prior to the jury deliberation, in presence of the related course professor. Without prejudice to the disciplinary processes at the University Faculty level, in case of fraud the student points for the related course are brought down to 0/20. The jury further can:

- *decide to cancel the examination session;*
- *decide to refuse the student access to both examination sessions of that academic year.*

Figure 2: Excerpt of the Exam Regulations concerning fraud.

HTTP operations to 1 megabyte. To change this limit run within your local repository

```
git config http.postBuffer *bytes*
```

where `*bytes*` is the maximum number of bytes permitted.

6. Your solution for phase 1 should be pushed to the repository *no later than 3 May 2019*. You get a penalty of -1/6 points for each day that your solution is delayed. Only the latest commit will be considered as the solution.

7. Sharing of code or reports between groups is not allowed. (Groups may, however, verbally discuss ideas on how to tackle the project).

8. Plagiarism, in the sense of copy-pasting from existing reports or books is not allowed. To avoid plagiarism, be sure to always quote your sources and indiate clearly if something has been copied verbatim. In case plagiarism is detected, students risk being punished according to article 34 of the exam regulations shown in Figure 2.

9. As stated above, this project assignment is done in groups of 2. All members of the group receive the same grade on the project! Therefore, in case that a group member feels that another group member does not do his/her share of the agreed work, please contact prof. Vansummeren immediately.