Big Data Management

Assignment

# Description

In this assignment your task is to compute the real-time views. You can use the same dataset as in previous assignments. You are required to use Apache Spark’s Streaming API to compute the real-time views. For storing these views you need to use the Apache Cassandra.

# Dataset

Transport Infrastructure Ireland (TII) (https://www.tii.ie) operates and maintains a network of tra and secondary road networks in Ireland. These tra data on diparameters. There are currently around counters active across the network. For an interactive view of the data they capture, go to the TII Trahttps://[www.](http://www/) nratrafficdata.ie). On this website, green dots display the individual tra counter locations around the country. Summary tra data infor- mation can be obtained from each site by clicking on the green dot. Upon clicking a dot, a pop-up window will appear that summarises the tra data and provides a link to more detailed data. You can click on the **list of sites** button to view description of each counter.

The tra hicle movements across the national road network and is made available publicly in its raw form, in order to provide researchers, public bodies, en- gineering companies, as well as the general public, with the opportunity to analyse and query the data independently for their own speci The row of each contains headers which describe each How- ever, the meaning of some of these may not be apparent to consumers. The following explains some of the less obvious column headers:

**cosit:** The unique identifier for the traﬃc counter device. In conjunction with the siteâĂŹs dataset, this can be used to determine the location and route of the counter, used to record the vehicle movement.

**lane:** The Id of the lane in which the movement was recorded, which is specific to each counter.

**straddlelane:** If a value is present, this indicates that the vehicle may have been changing lanes as it passed over the counter.

**class/classname:** This indicates the category of vehicle that was recorded e.g. car, bus, etc.

**length:** The approximate length of the vehicle recorded.

**headway:** The approximate distance between the front of the recorded vehicle and the vehicle behind.

**gap:** The approximate distance between the rear of the vehicle and the front of the vehicle behind.

**weight:** This is available on (Weigh-in-Motion) WIM sites only and indicates the approximate weight of the vehicle.

**temperature:** If available, this indicates the approximate surface temperature of the road at the location of the device.

**numberofaxles:** This is available on WIM sites only and indicates the number of axles detected for the vehicle.

**axleweights:** This is available on WIM sites only and expresses as an array of real numbers, the weight over each axel in order.

**axlespacing:** This is available on WIM sites only and expresses as an array of real numbers, the distance between each of the axles.

# Questions

To answer the questions below you must use Apache Spark’s Streaming API. This time we are interested in real-time processing of the tra dataset. The results should be persisted in the Cassandra storage (use the append option).

To emulate a live-stream of the tra counter dataset, you are re- quired to write a separate Python script that reads records (of

each counter site - ignore the test site) every 10 seconds from traffic

counter data and stores them as separate (countdata1, countdata2, countdata3, etc.) in the streaming directory on which your application is listening.

Prepare the streaming application to read the data streams from the streaming directory using a batch length of 5 seconds.

Do the following streaming computations (every 5 seconds):

* + - Show total number of counts (on each site of M by vehicle class.
    - Compute the average speed (on each site on M by vehicle class.
    - Find the top busiest counter sites on M50
    - Find total number of counts for HGVs on M50

Store the results of streaming computations de above:

* + - Prepare Cassandra data structures to store the results.
    - Prepare code for writing the results into the Cassandra tables.