Week 8: Data Streaming Management Systems Assignment

Introduction:

The goal of this assignment is to illustrate the different concepts and ideas about processing of continuous queries, using techniques such as sliding windows or sampling.

Instructions will be provided to connect to the Twitter API and continuously read a stream of tweets associated to a certain account.

The students will be required to implement a script that performs continuous update of some fixed Continuous Queries. We will define the queries in natural language, so the students will also have to translate them into some CQ language (although they are not required to parse the language, their program will just implement continuous update of their interpretation of the queries).

Preliminary Readings:

1. For example: Models and Issues in Data Stream Systems, by Babcock, Babu, Datar, Motwani, & Widom. (Section 6 is not needed.)

Research Questions

1. Think about the following two situations:
    a. Analyze the data of recent power usage statistics reported to a power station and adjust the power generate rate if necessary.
    b. A school needs to know data about their students, and query about courses, age, and instructors.

    Which system would you choose for each situation? If you choose DSMS, explain in detail how you would implement it.
2. Explain three of the most typical strategies for CQ processing, and for each of them, think of scenarios where it would be more convenient to apply it.
3. Mention the main characteristics of a Query Scheduler in a Data Stream Management System. What do you think is the biggest problem of the scheduling in comparison with a Data Base Management System?

Lab assignment:
Continuous Querying over Twitter Stream

Readings:

J.B. Hester (2014), "Creating a Python Script for Twitter Search"
http://coding2day.com/TwitterPython.pdf

Requirements

a. Python 2.7 or newer
b. Libraries
c. Auth and consumer key of a twitter account in https://developer.twitter.com/

Development

1. Write queries in CQL (objective is to understand the queries, but the script isn't in SQL).
2. Implement a query processing script in python over a Tweeter Data Stream.
3. Give the answer of the **queries** below.
4. Think of four queries over the Twitter Data Stream. Assume different flow rates and computing time.
5. Write your conclusions.

Queries:

- percentage of tweets mentioning "Yes" during 2 minutes, from all tweets, without restrictions.
- (sliding windows) Use sliding windows, and calculate the percentage of tweets that mention "Yes". Use a window size of 300 tweets.
- (batching) percentage of tweets mentioning "No" during 2 minutes. Assume that the counting function has a very slow rate, use buffers (length 100) for the elements and compute the query answer using each tweet once.
- (sampling) percentage of tweets mentioning "Hi" during 2 minutes. Assume that the update function is slow. Update the list with a sample of the elements. (e.g., one in one hundred).

Useful Resources:

1. https://developer.twitter.com/
   https://developer.twitter.com/en/docs/tweets
   https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets
2. https://code.google.com/p/python-twitter/ (https://github.com/bear/python-twitter)