CS594: Python Programming Lab

Take Home Assignment - 5 (2 Questions, 100 Points)
Submission Dead Line: 28-Oct-2019 23:59 Hours Pages: 3

IIT Guwahati

14 Oct 2019 (Mon)

Question 1: (0 points)
Reading the following

Kafka `http://kafka.apache.org/`

Kafka Python API `https://kafka-python.readthedocs.io/en/master/apidoc/modules.html`

Book Source `Kafka The Definitive Guide`

Question 2: (100 points)
The objective is to handle *data streams* and process the data in real time.

Problem Setting Assume that *all* the 48 cricket world cup matches are happening *simultaneously*.

- Every commentator (`producer`) is writing the commentary (*message*) to `Kafka server`. Each commentary line is prefixed with `match_id` and `innings_id`.
- The `match_id` serve as *topics*
- The messages are not ordered in any specified manner. That is the stream is shown in figure 1. In this figure, text commentary from all the matches appears. In the `Kafka` server, you will not get complete text commentary of `match_id = 1234` followed by complete text commentary of `match_id = 1235` etc. The commentary will be interleaved in nature.
- There should be 48 producers one corresponding to each `match_id`.
- There should be 48 consumers one corresponding to each `match_id`.
- Consumers subscribe to the commentary based on the *topic*. In this case `match_id`.

1. Write a Python program for producing the commentary messages to the Kafka Server.
2. Write the messages to Kafka server such that the messages are interleaved.
3. Write a Python program for consuming the commentary messages from the Kafka Server.
4. The consumers should listen to a specified topic (specified match) and collect all commentaries pertaining to that match
5. Your task is to *compute the scorecard* from the consumers topics. Output scorecard should be *identical in format* as given at the `SCORECARD` link. Store the scorecard in an outp with the name `roll-number-match-id-scorecard-computed.txt`.
6. Scorecard should contain the following three sections:

(a) Batsmen
(b) Fall of wickets
(c) Bowling

7. You have to compute scorecard for all the 48 matches.
8. You `should NOT use` the following information present in the text commentary (and you should not write them into the Kafka server)
   - Summary lines written just after the wicket taken by a bowler such as in the URL `https://www.espncricinfo.com/series/8039/commentary/1144483/england-vs-south-africa-1st-match-icc-cricket-world-cup-2019?innings=2&filter=full`
     `Over 25.5` use of lines of the form
     JP Duminy c Stokes b Ali 8 (12m 11b 1x4 0x6) SR: 72.72
   - End of over summaries such as in the same URL above, at the end of 26 overs summary available in a grey box presented as
     END OF OVER:26 | 5 Runs 1 Wkt | SA: 142/4 (170 runs required from 24 overs, RR: 5.46, RRR: 7.08)
   - Use of the above information to compute scorecard leads to penalization of 50% of marks Batting Section marks. That is you will loose 10 marks for using this information.
9. **You are permitted** to use the above information to extract number of minutes played by a batsman as this information cannot be `computed` from text commentary alone.
10. You must use regular expressions for processing each line of the text commentary.
11. You must use two classes one for constructing batting part of the scorecard; other for the bowling part of the score card.

Instructions File Naming Convention Create a directory with your roll number. Inside this directory, place all the above python programs and input files. Prefix the file name with your roll number followed by "_" followed by question number followed by ".py". Example: 194161000_q1.py.

README.txt Write a short notes on sequence of steps involved to run the your programs. Include what is the input for the program (with an example) and what will be the output from the program (with an example).

tar gzip Create (roll number).tar.gz file using the above directory. This directory must contain the following:

Q2. The input files prepared for Q2
Q2. Python program solution for Q2
README Instructions to run your program must be placed in README.txt file.

Submission Email the above tar gzip file to the CS594 TA `vaibhav18@iitg.ac.in` as per the above given dead line

Copying You should avoid indulging in copying. Every submission will be subject to software similarity using the tool `Measure of Software Similarity`

available at `https://theory.stanford.edu/~aiken/moss/`. Two submissions having similarity score equal to or more than 40.0% will be declared copied. If you are found involved in copying act, your name will be referred to disciplinary committee. Therefore you are requested to place individual efforts and avoid copying.

Marking Scheme Your implementation will be evaluated as described below.

- **Q2. 35 Marks** Writing the producer for each match to the Kafka server.
- **Q2. 35 Marks** Writing the consumer for each match from the Kafka server by listening to a particular topic.
- **Q2. 10 Marks** Writing message in non-interleaving fashion
- **Q2. 20 Marks** Computing the score cards from each of the consumers.
- **Q2. 20 Bonus Marks** If you demonstrate the procedures and consumers in a distributed environment setting.