Big Data Management

Assignment 3

Description

In this assignment your task is to compute the real-time views. 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. You will be using the CRAN package download logs (`http://cran-logs.rstudio.com`). These log files contain all hits to `http://cran.rstudio.com` mirror related to downloads of the R packages. The raw log files have been parsed into CSV and anonymised. Since these logs contain massive amount of data (from 2012 to date), we will only be using a recent one which represent the logs for 31st of October, 2021. This log file are available at:

`http://cran-logs.rstudio.com/2021/2021-10-31`

The package download logs contain data about the following variables:

```
date: Download date
time: Download time (in UTC)
size: Package size (in bytes)
r_version: Version of R used to download package
r_arch: Processor architecture (i386 = 32 bit, x86_64 = 64 bit)
r_os: Operating System (darwin9.8.0 = mac, mingw32 = windows)
package: Name of the package downloaded
country: Two letter ISO country code
ip_id: A daily unique id assigned to each IP address
```

Setting Up

Follow the `Getting Started with Spark Streaming` document provided on Moodle. For help Spark Streaming API and various operations and transformations that you can apply on DStreams see the following links:

```
http://spark.apache.org/docs/latest/api/python/pyspark.streaming.html
https://spark.apache.org/docs/latest/streaming-programming-guide.html
```

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 CRAN package download logs. The results should be persisted in the Cassandra storage (use the `append` option).

1. To emulate a live-stream of the download logs, you are required to write a separate Python script that reads 1000 lines every 3 seconds from the log file and stores them as separate files (`log1, log2, log3, etc.`) in the streaming directory on which your application is listening. `(30 marks)`
2. Prepare the streaming application to read the data streams from the streaming directory using a batch length of 3 seconds. `(10 marks)`

Define the following streaming computations (every 3 seconds):

1. To calculate the number of downloads of each package. `(10 marks)`
2. To find the top most downloaded package. `(10 marks)`
3. To find the top 5 countries along with number of downloads. `(10 marks)`
4. To find total number of downloads for `ggplot2` package. `(10 marks)`

Store the results of streaming computations defined above:

1. Prepare Cassandra data structures to store the results. `(5 marks)`
2. Prepare code for writing the results into the Cassandra tables. `(15 marks)`

Submission

- Create a PDF document that contains the code you used to answer each query AND 2-3 of screenshots of the results stored in Cassandra (after running your streaming application for 2 minutes for each question).
- Acceptable code file format: Python notebook - name it `assignment3.ipynb`. The notebook should be exported as iPython Notebook with *.ipynb extension. If the code in your notebook does not run, it will result in 20% penalty.
- Zip both files: the PDF and the Python notebook. Submit the zip file on Moodle before the deadline.
- Do not submit work thats not your own and do not let others copy work that is your own. Both Copyier and Copyee will get ZERO marks.