CS4830: Big Data Lab

Divya K Raman, EE15B085

Week 8 Report

2 April 2019

Answer 1

Discretized Stream or DStream is the basic abstraction provided by Spark Streaming. It represents a continuous stream of data, either the input data stream received from source, or the processed data stream generated by transforming the input stream. This command returns a DF. Spark signifies that the language is spark. .readStream signifies that data needs to be read from the stream. jsonSchema declares the input streaming data to be a json file. The input path of the json file is then set and maxFilesPerTrigger indicates the maximum number of new files to be considered in every trigger.

References:

https://spark.apache.org/docs/2.2.0/streaming-programming-guide.html#discretized-streams-dstreams

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Answer 2

Modifying the code to filter out all objects with action = “close”:

streamingCountsDF1 = streamingCountsDF.where(streamingCountsDF['action']=='Close')

Answer 3

In this command, we query the dataframe and write the result using writeStream. .format("memory") signifies that it is a Memory sink. The output is stored in memory as an in-memory table. The streaming computation is started using start(). Query name is an optional parameter to specify a unique name of the query for identification. Output mode specifies what gets written to the output sink.

Parameter values for output mode: append, complete, update

Parameter values for format: parquet, console, memory,

Reference:

https://spark.apache.org/docs/2.2.0/structured-streaming-programming-guide.html

Answer 4

Spark Streaming is based on DStream which is represented by a continuous series of RDDs, RDDs are Spark's abstraction of an immutable, distributed dataset. Spark Streaming is difficult and inconsistent. It is hard to build because streaming pipelines supporting delivery policies like exactly once guarantee and handling data arrival in late or fault tolerance. It is inconsistent as API used to generate batch processing (RDD, Dataset) is different than the API of streaming processing (DStream).

Spark Structured Streaming be understood as an unbounded table, growing with new incoming data, i.e. can be thought as stream processing built on Spark SQL. More concretely, structured streaming brought some new concepts to Spark. Structured streaming focuses on the concept of exactly once guarentee. It means that data is processed only once and output doesn't contain duplicates. One of the observed problems with DStream streaming was processing order, i.e the case when data generated earlier was processed after later generated data. Structured streaming handles this problem with a concept called event time that, under some conditions, allows to correctly aggregate late data in processing pipelines.

Our code uses spark streaming.

Reference: https://stackoverflow.com/questions/49290521/what-is-the-difference-between-spark-structured-streaming-and-dstreams

Answer 5

There can be two kinds of sources based on their reliability. Reliable Sources allow the transferred data to be acknowledged. If the system receiving data from these reliable sources acknowledges the received data correctly, it can be ensured that no data will be lost due to any kind of failure.

Kafka and flume are reliable sources. JSON is not.

Reference: https://spark.apache.org/docs/latest/streaming-programming-guide.html