speed-processing

December 23, 2020

[1]:
```
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
from pyspark.sql.functions import *
from pyspark.sql import functions as F
import pandas as pd

from pyspark.sql import functions as sf
import json
import time
from pyspark.sql.functions import col
import yaml
```

[2]:
```
ss = SparkSession.Builder() \
     .appName("SparkSpeedStreamingKafka") \
     .master("spark://speed-processing-spark-master:7077") \
     .config("spark.jars", "./jars/spark-streaming-kafka-0-10-assembly_2.11-2.4.
↪1.jar,./jars/kafka-clients-0.10.1.0.jar,./jars/spark-sql-kafka-0-10_2.11-2.4.
↪1.jar") \
     .config("spark.sql.warehouse.dir", "hdfs://namenode:9000/") \
     .getOrCreate()
```

[3]:
```
df = ss \
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", "kafka-broker-1:9093,kafka-broker-2:9093")␣
↪\
  .option("partition.assignment.strategy", "none") \
  .option("subscribe", "trips") \
  .load()
```

[4]:
```
import random

def transform_window(s):
    """
    s = Row(start=datetime.datetime(2020, 12, 21, 17, 9, 30), end=datetime.
↪datetime(2020, 12, 21, 17, 9, 40))
    """
    return str(int(s.end.timestamp()))

def transform_count(s):
    """
    s = 941
    """
    return str(s)
udf_transform_window = udf(transform_window)
udf_transform_count = udf(transform_count)
```

[ ]:
```
query = df.withWatermark("timestamp", "20 seconds") \
        .groupBy(window("timestamp", "10 seconds", "10 seconds")) \
        .count() \
        .withColumn("count", udf_transform_count("count")) \
        .withColumn("window", udf_transform_window("window")) \
        .withColumn('value', sf.concat(sf.col('window'),sf.lit('_'), sf.
↪col('count'))) \
        .writeStream \
        .format("kafka") \
        .option("kafka.bootstrap.servers", "kafka-broker-1:9093,kafka-broker-2:
↪9093") \
        .option("topic", "real-time-statistic") \
        .option("checkpointLocation", "/tmp/checkpoint") \
        .outputMode("append") \
        .option("truncate", False) \
        .start()
query.awaitTermination()
```

[ ]: