Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Методи і засоби обробки великих даних

Лабораторна робота №6

«Об’єднання потоків в Spark Streaming»

Виконав:

студент групи ІС-12мп

Коноплянка Д. С.

Перевірила:

Тимофєєва Ю. С.

Київ 2021 р.

1

import fs from 'fs'

import { v4 as uuidv4 } from 'uuid'

import randomWords from 'random-words'

const type = {

hadoop: 'hadoop',

spark: 'spark',

kafka: 'kafka'

}

const keys = Object.keys(type)

const delayed = () => {

return new Promise((resolve) => {

setTimeout(() => {

const data = fs.readFileSync("./data/data.json")

const myObject = JSON.parse(data)

const prop = keys[Math.floor(Math.random() \* keys.length)]

const newData = {

id: uuidv4(),

text: randomWords(),

type: prop,

createDate: Date.now()

};

myObject.push(newData)

const newJson = JSON.stringify(myObject);

fs.writeFile("./data/data.json", newJson, (err) => {

if (err) throw err

console.log("New data added")

console.log(newData)

});

resolve()

}, 1000);

})

}

for (; ; ) {

await delayed()

}

val data = Seq(("hadoop", "1"), ("spark", "2"), ("kafka", "3"))

val rdd = spark.sparkContext.parallelize(data)

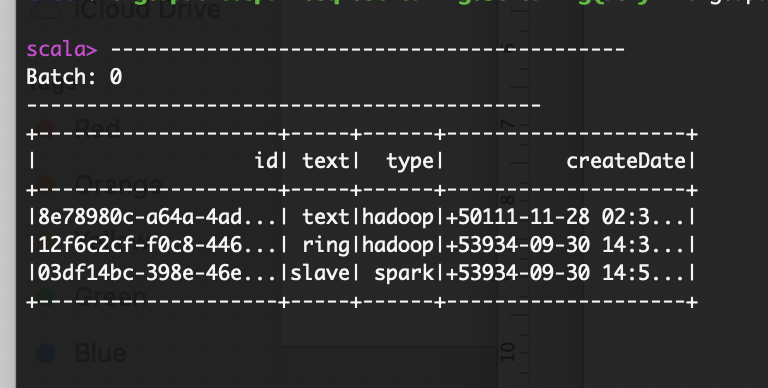
val dfF = rdd.toDF("type", "value")

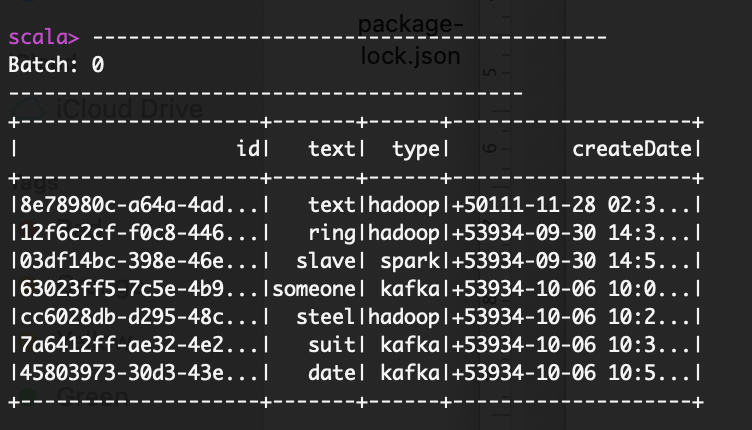
import org.apache.spark.sql.types.{StructType, StructField, StringType, IntegerType}

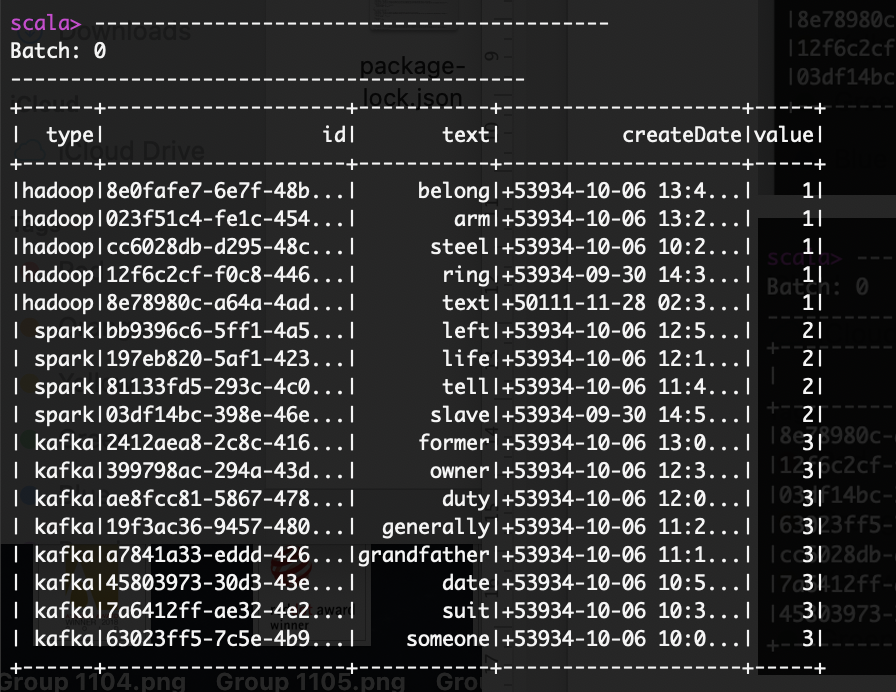
val schema = new StructType().add("id", "string").add("text", "string").add("type", "string").add("createDate", "timestamp")

val csv = spark.readStream.schema(schema).json("/Users/kanopa/Documents/education/kpi/5course/BigData/lab6/data")

val check = csv.writeStream.outputMode("update").format("console").start()







val joined = csv.join(df, "type")

joined.writeStream.outputMode("update").format("console").start()

joined.writeStream.format("json").option("path", "/Users/kanopa/Desktop").option("checkpointLocation", "/Users/kanopa/Desktop").start()

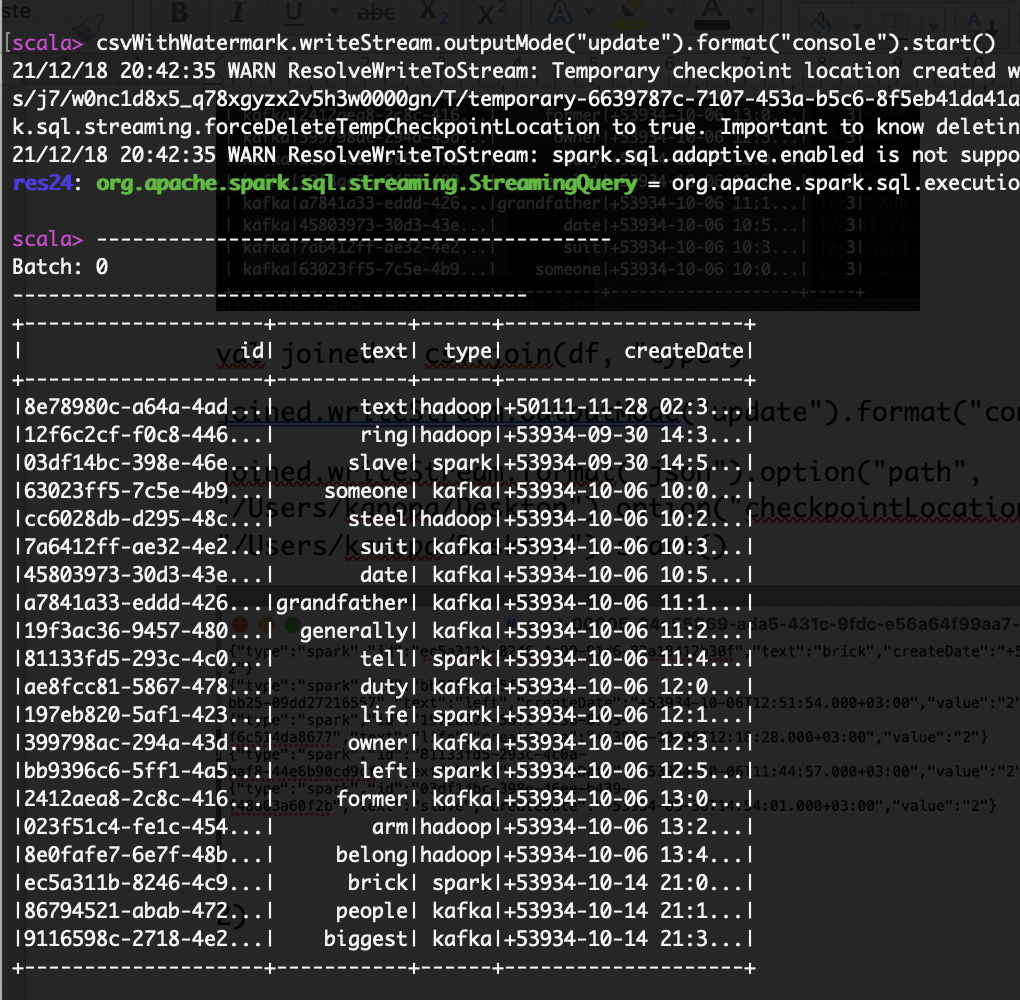


2)

val csvWithWatermark = csv.withWatermark("createDate", "2 hours")

val csv2WithWatermark = csv2.withWatermark("createDate2", "3 hours")

csvWithWatermark.join(csv2WithWatermark, expr(""" createDate > createDate2 """))



csvWithWatermark.writeStream.format("json").option("path", "/Users/kanopa/Desktop").option("checkpointLocation", "/Users/kanopa/Desktop").start()

