Paddling Up the Stream: Lessons Learned using Apache Spark Streaming

Miklos Christine
mwc@databricks.com
Oct 2016

databricks

$ whoami

- Previously Systems Engineer @ Cloudera
- Deep Knowledge of Big Data Stack
- Apache Spark Enthusiast
- Solutions Architect @ Databricks!

Who is Databricks

Why Us

- Created Apache Spark to enable big data use cases with a single engine.
- Contributes heavily to Spark's codebase

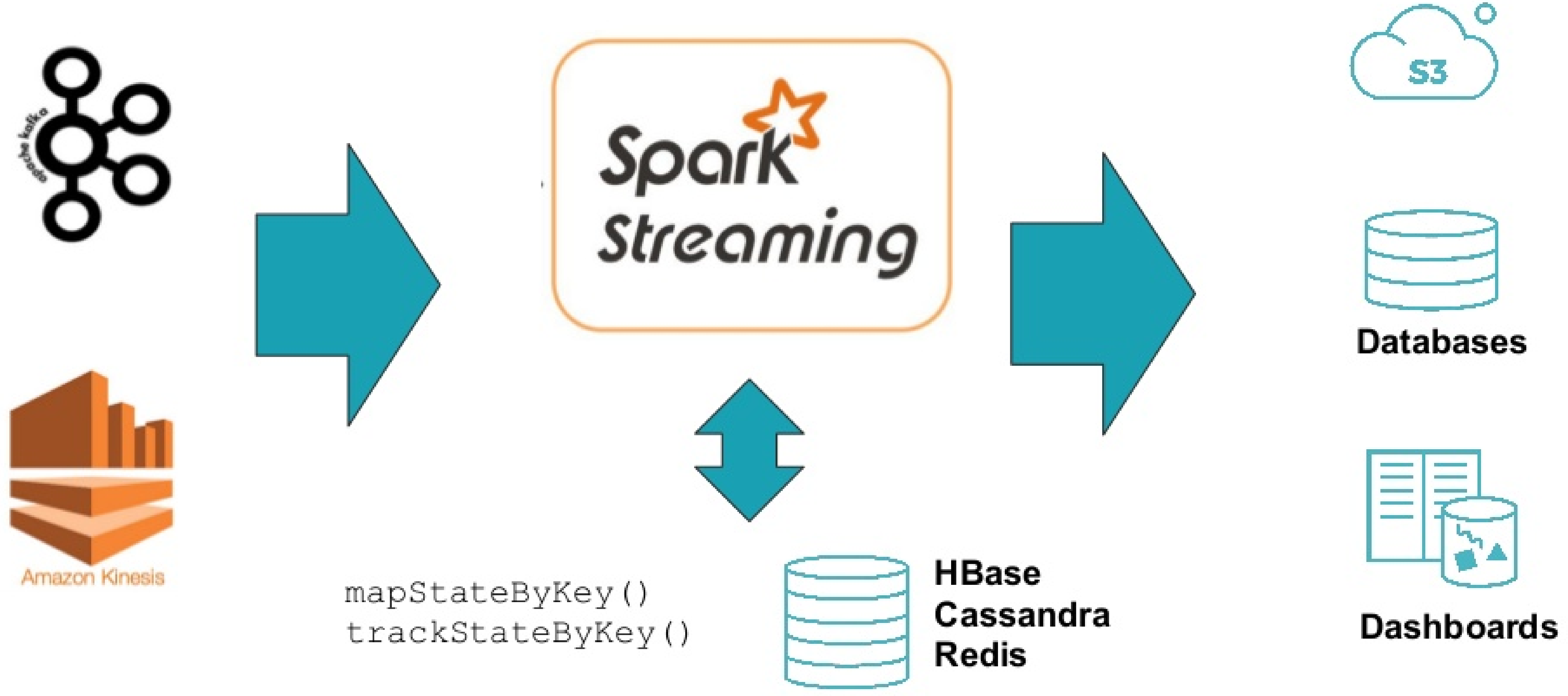
Our Product

- Bring Spark to the enterprise: The just-in-time data platform.
- Fully managed platform powered by Apache Spark.
- A unified solution for data science and engineering teams.

Overview

- Architecture Decisions
- Spark Streaming vs Structure Streaming
- Top 5 Support Issues

Design Decisions



```
mapStateByKey()
trackStateByKey()
```

Streaming Options

DStreams - 1.5+

- RDD based APIs
- Supports Kafka and Kinesis
- Production deployments

Structured Streaming - 2.0+

- Streaming API built on top Datasets
- Experimental [alpha]
- Supports limited sinks / sources
- Continuous applications

Issue 1: Type Mismatch

- Stacktrace

```
error: object clientlibrary is not a member of package
com.amazonaws.services.kinesis
import
com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream

InitialPositionInStream.LATEST, batchInterval, StorageLevel.MEMORY_AND_DISK_2,
        ^
<driver>:72: error: type mismatch;
 found   : (org.apache.spark.rdd.RDD[Array[Byte]],
org.apache.spark.streaming.Time) => Unit
 required: (org.apache.spark.rdd.RDD[Nothing], org.apache.spark.streaming.Time) =>
Unit
    unionStreams.foreachRDD { (rdd:RDD[Array[Byte]], time: Time) =>
```

Issue 1: Type Mismatch

- **Solution**: Libraries libraries libraries!
- Ensure the correct libraries are in use

Connector jar (mvn coordinate)

```
org.apache.spark:spark-streaming-kinesis-asl_2.11:2.0.0
```

Kinesis jar

```
com.amazonaws:amazon-kinesis-client:1.4.0
```

Ref:

https://mvnrepository.com/artifact/org.apache.spark/spark-streaming-kinesis-asl_2.11/2.0.0

https://mvnrepository.com/artifact/com.amazonaws/amazon-kinesis-client/1.4.0

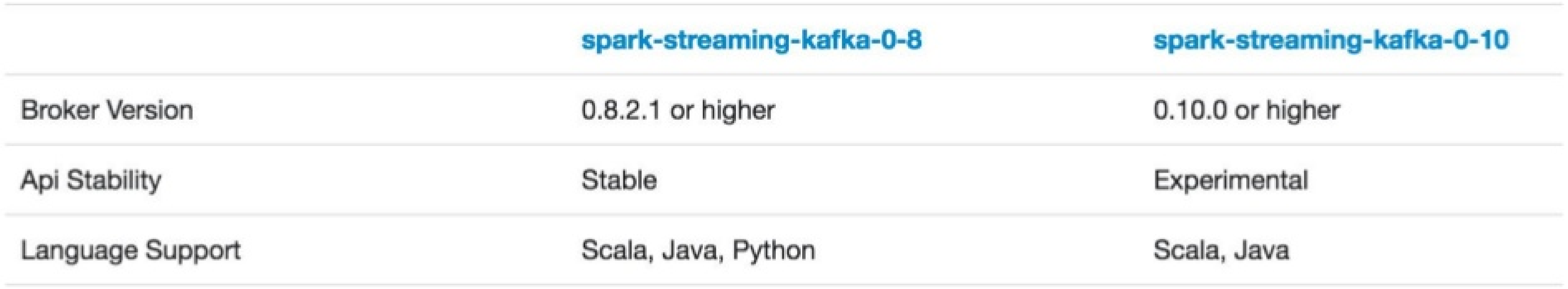
Issue 2: Couldn’t find leader offsets

Stacktrace:

```
org.apache.spark.SparkException: java.nio.channels.ClosedChannelException
org.apache.spark.SparkException: Couldn't find leader offsets for Set([databricks,0], [databricks,1])
at org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
at org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
at scala.util.Either.fold(Either.scala:97)
```

Problem: Spark 2.0 + 0.8 Kafka connector ⇒ Kafka 0.10 Cluster

Issue 2: Couldn’t find leader offsets



	spark-streaming-kafka-0-8	spark-streaming-kafka-0-10
Broker Version	0.8.2.1 or higher	0.10.0 or higher
Api Stability	Stable	Experimental
Language Support	Scala, Java, Python	Scala, Java

Solution

- Documentation released after code was available
  - Docs 2.0.1 / Code 2.0.0
- `spark-streaming-kafka-0-8-assembly_2.11`
  - `spark-streaming-kafka-{KAFKA_VER}-assembly_{SCALA_VER}`

Issue 3: toDF not member of RDD

- **Stacktrace**

```
error: value toDF is not a member of org.apache.spark.rdd.RDD[(java.sql.Timestamp,
String, String, Int, Int, Option[Array[String]], Option[String], Option[Double],
Option[Double])]

       val df = temp.toDF("createdAt", "user", "tweet", "favCount",
"retweetCount", "hashtags", "countryCode", "lat", "lon")
```

Issue 3: toDF not member of RDD

- **Solution**

```
val _spark = org.apache.spark.sql.SparkSession.builder().getOrCreate()

val _sqlContext = _spark.sqlContext

import _sqlContext.implicits._
```

- Last line is the important one.
  - Import is in most examples, but the error message isn’t clear why it’s failing.

Issue 4: Task Not Serializable

- **Stacktrace**

```
org.apache.spark.SparkContext
Serialization stack:
     - object not serializable (class: org.apache.spark.SparkContext, value:
org.apache.spark.SparkContext@6c7a65d3)
     - field (class: com.databricks.example.SendTweetsToKinesis, name: _sc, type:
class org.apache.spark.SparkContext)
     - object (class com.databricks.example.SendTweetsToKinesis,
com.databricks.example.SendTweetsToKinesis@510c02d6)
     - field (class:
com.databricks.example.SendTweetsToKinesis$$anonfun$creatingFunc$1, name: $outer,
type: class com.databricks.example.SendTweetsToKinesis)
```

Issue 4: Task Not Serializable

- **Solution**

```
val _spark = org.apache.spark.sql.SparkSession.builder().getOrCreate()
val _sc = _spark.sparkContext
val _sqlContext = _spark.sqlContext
import _sqlContext.implicits._
SparkSession.setActiveSession(_spark)
```

- Get SparkSession inside the streaming create functions and set the active session

Issue 5: Push JSON Records

- Question: How do I efficiently push JSON records to Kinesis / Kafka?

- Solution:

```
val df = temp.toDF("createdAt", "user", "tweet", "favCount", "retweetCount",
"hashtags", "countryCode", "lat", "lon")

json_rdd = df.toJSON.rdd

json_rdd.foreachPartition ( partition => {

     // Send records to Kinesis / Kafka

     })
```

Bonus: Performance

- Read through the performance page in the upstream docs. Link below.
- Adaptive input rates can be configured:

```
spark.streaming.receiver.maxRate for receivers

spark.streaming.kafka.maxRatePerPartition

spark.streaming.backpressure.enabled true
```

Ref:

http://spark.apache.org/docs/latest/streaming-programming-guide.html#setting-the-right-batch-int
erval

Thank you.

Try Databricks Today!

https://databricks.com/try-databricks