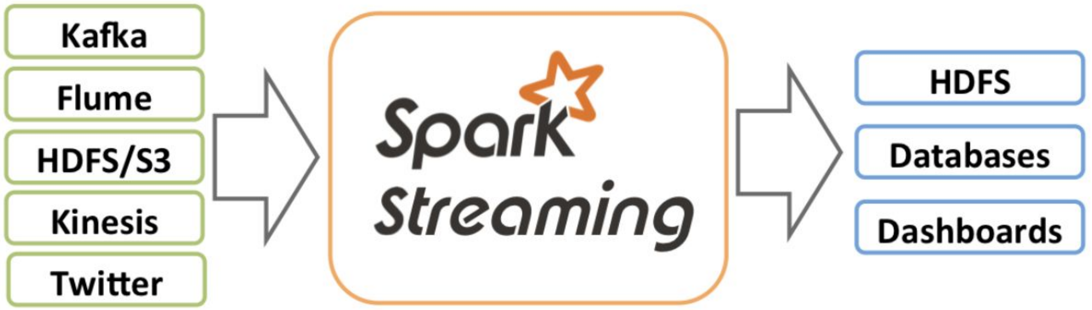
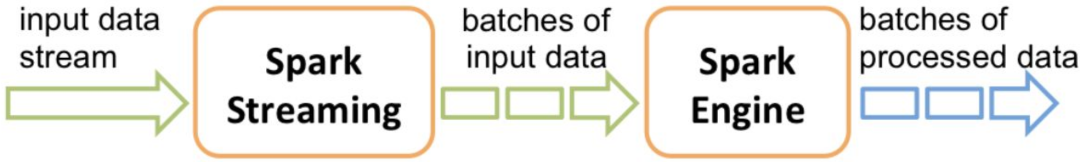
Agenda

1. Spark Streaming
2. Assignment 9 Overview
3. Demo

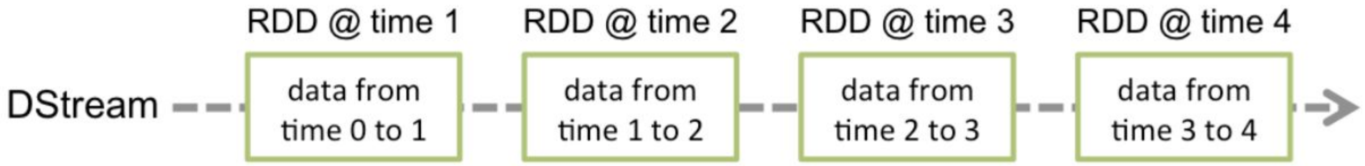
Streaming with Spark



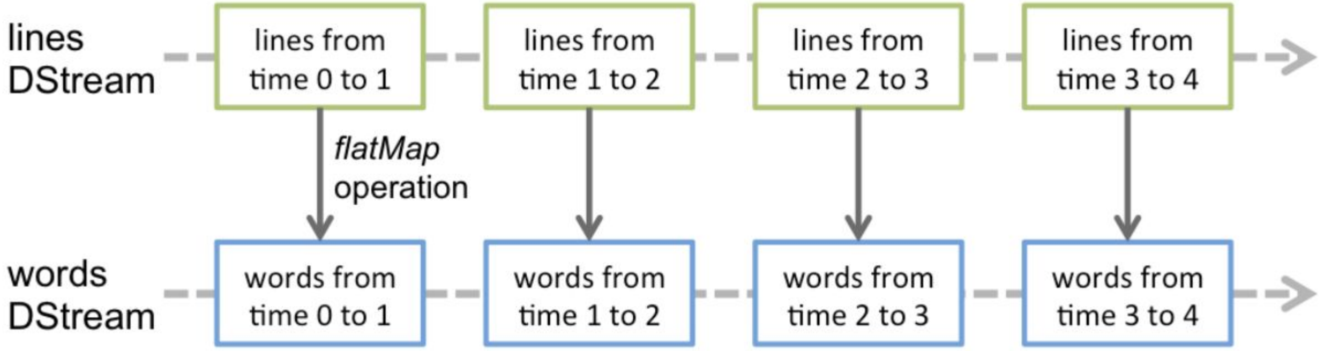


https://spark.apache.org/docs/latest/streaming-programming-guide.html

Dstream - discretized stream

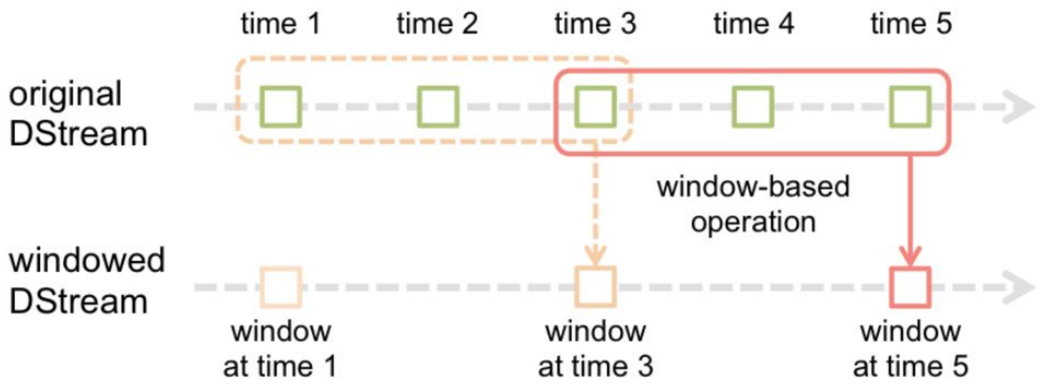


Wordcount example



Reference: https://spark.apache.org/docs/latest/streaming-programming-guide.html#transformations-on-dstreams

Windows operations



- *window length* - The duration of the window.
- *sliding interval* - The interval at which the window operation is performed.

Reference: https://spark.apache.org/docs/latest/streaming-programming-guide.html#transformations-on-dstreams

Python env setup

```
from pyspark import SparkContext, SparkConf
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
```

```
--packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.0.0
```

Advanced Sources
Python API As of Spark 2.4.4, out of these sources, Kafka, Kinesis and Flume are available in the Python API.

Java env setup

1. Sample maven config for Kafka Producer

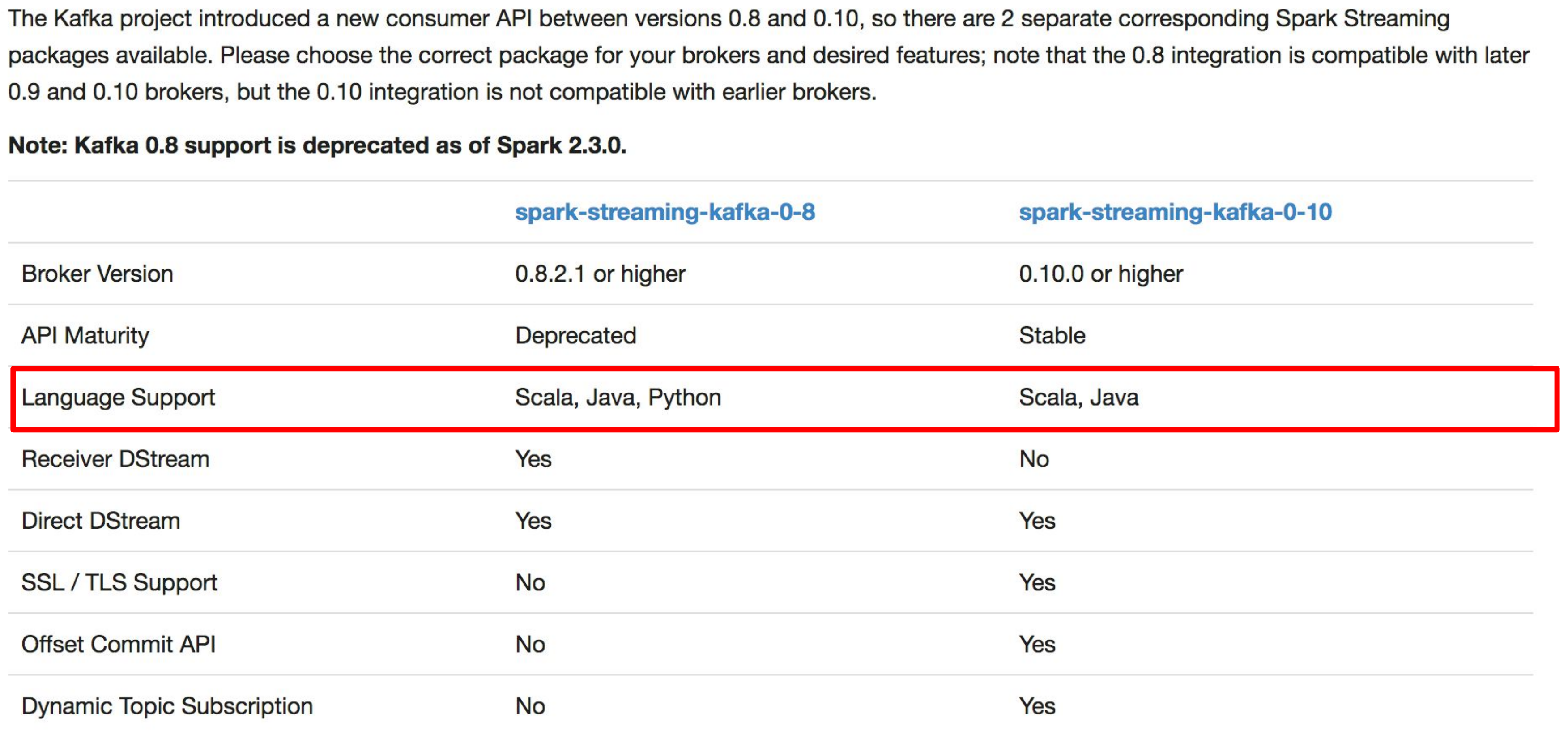
```
 <dependency>
                 <groupId>org.apache.kafka</groupId>
                 <artifactId>kafka-clients</artifactId>
                 <version>2.0.0</version>
</dependency>
```

2. Sample maven config for Spark Job

```
<dependency>
      <groupId>org.apache.spark</groupId>
       <artifactId>spark-streaming_2.12</artifactId>
       <version>2.4.4</version>
       <scope>provided</scope>
</dependency>
 <dependency>
        <groupId>org.apache.spark</groupId>
         <artifactId>spark-streaming-kafka-0-10_2.12</artifactId>
         <version>2.4.4</version>
 </dependency>
```

3. Either generate the JAR with all dependencies or use --package option to define the dependency

Kafka Connecter



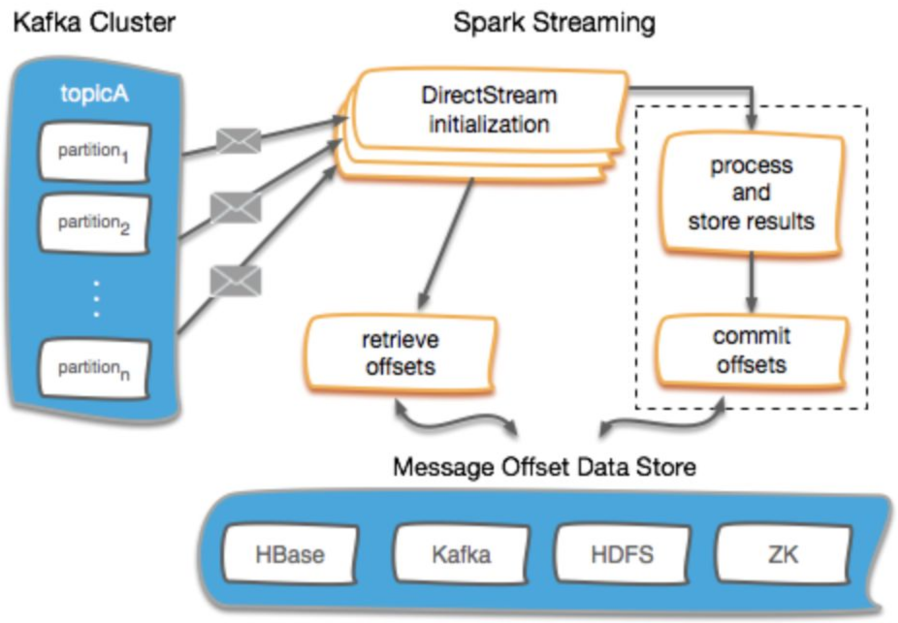
The Kafka project introduced a new consumer API between versions 0.8 and 0.10, so there are 2 separate corresponding Spark Streaming packages available. Please choose the correct package for your brokers and desired features; note that the 0.8 integration is compatible with later 0.9 and 0.10 brokers, but the 0.10 integration is not compatible with earlier brokers.

Note: Kafka 0.8 support is deprecated as of Spark 2.3.0.

	spark-streaming-kafka-0-8	spark-streaming-kafka-0-10
Broker Version	0.8.2.1 or higher	0.10.0 or higher
API Maturity	Deprecated	Stable
Language Support	Scala, Java, Python	Scala, Java
Receiver DStream	Yes	No
Direct DStream	Yes	Yes
SSL / TLS Support	No	Yes
Offset Commit API	No	Yes
Dynamic Topic Subscription	No	Yes

https://spark.apache.org/docs/latest/streaming-kafka-integration.htm

Offsets management - important for Prod deployment



[figure 1 – high-level flow for managing offsets]

https://blog.cloudera.com/offset-management-for-apache-kafka-with-apache-spark-streaming/

Kafka and Spark Streaming Demo

SetupKafka-AWS-PublicSubnets.json - Cloudformation script to setup Kafka in AWS.

Useful commands

Run java spark job

```
spark-submit --class edu.harvard.e88.lab9spark.SparkStreamConsumer lab9spark-0.0.1-SNAPSHOT-jar-with-dependencies.jar
mytopic  kafkabrokerIP:9092
```

Run Python spark job

```
spark-submit --packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.0.0 SparkStreamingConsumer.py kafkabrokerIP:9092
mytopic
```

Run Kafka Producer

```
java -cp lab9-0.0.1-SNAPSHOT-jar-with-dependencies.jar edu.harvard.e88.lab9.WeblogsKafkaProducer mytopic kafkabrokerIP:9092
```

Create Kafka Topic

```
./kafka-topics.sh --zookeeper ZookeeperIP:2181 --create --topic mytopic --partitions 2 --replication-factor 1
```

UpdateStateByKey

Additional References

https://blog.cloudera.com/offset-management-for-apache-kafka-with-apache-spark-streaming/

https://aws.amazon.com/blogs/big-data/real-time-stream-processing-using-apache-spark-streaming-and-apache-kafka-on-aws/