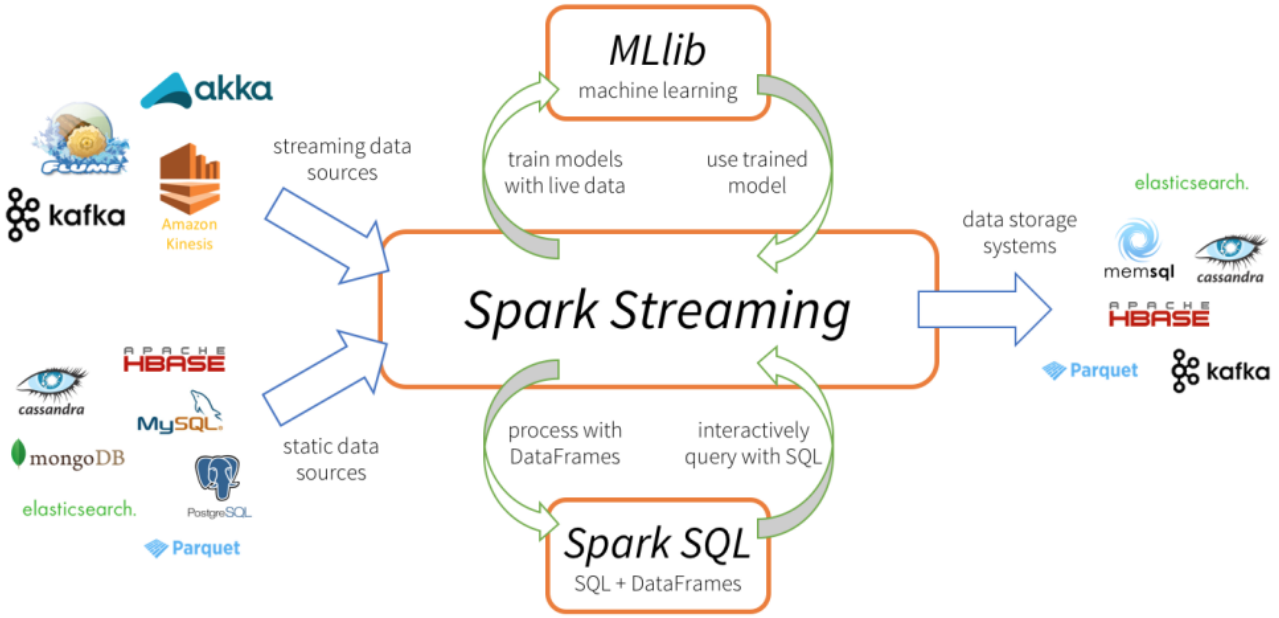
PROCESSING OF BIG DATA

SPARK

SESSION-4


akka
kafka
Amazon Kinesis
streaming data sources
cassandra
APACHE HBASE
MySQL
mongoDB
PostgreSQL
elasticsearch.
Parquet
static data sources
MLlib
machine learning
train models with live data
use trained model
Spark Streaming
process with DataFrames
interactively query with SQL
Spark SQL
SQL + DataFrames
data storage systems
elasticsearch.
memsql
cassandra
APACHE HBASE
Parquet
kafka

Agenda

- Unstructured Streaming
- Structured Streaming

Data Holder

- *Discretized stream* or ***DStream***

- Internally, a DStream is represented as a sequence of RDDs

- lines = ssc.socketTextStream(“localhost”, 9999)

DStream API

- map
- flatMap
- filter
- repartition
- union
- count
- countByValue
- reduceByKey
- join
- **updateStateByKey**
- **transform**

UpdateStateByKey

- Refer to the following code for better understanding:

https://github.com/apache/spark/blob/v2.4.5/examples/src/main/python/streaming/stateful_network_wordcount.py

Transform

- Allows arbitrary RDD-to-RDD functions to be applied on a DStream

- Used to apply any RDD operation that is not exposed in the DStream API

- E.g.: Join an RDD with DStream

Transform: Cont…

lines = ssc.socketTextStream(“localhost”, 9999)

#Generate key value pairs **DStream**

keyVal = lines.map(lambda x: (x.split(",")[0],x))

#Read product **RDD** in memory

productRdd = sc.textFile("product.csv")

product = productRdd.map(lambda x: (x.split(",")[0],x))

joined = keyVal.transform(lambda rdd: rdd.join(product))

Checkpoint

- Metadata: Saving of the information defining the streaming computation to fault-tolerant storage like HDFS
- Stores*: Configuration, DStream operations, Incomplete batches*

- Data: Saving of the generated RDDs to reliable storage
- Stores: intermediate RDDs of stateful transformations

Windowing operation

from pyspark.sql.functions import window

```
df.groupBy(window(df.timestamp,
        "2 minutes", "1 minutes"),
        product_id, name)
        .agg(_sum("qty"), _sum("amt"))
```

Unstructured Streaming

- Source
  - Socket: exercise 1
  - File: exercise 2
  - Kafka: exercise 3

- Sink
  - File: exercise 4
  - Database: exercise 5

Unstructured Streaming: Cont..

- ETL on streaming data
  - Transform operation: exercise 6
- Checkpoint for restartability
  - Metadata checkpoint: exercise 7
  - Data checkpoint: self study
- Social media feed processing
  - Twitter feed analysis: self study

Structured Streaming

- Source
  - Socket: exercise 8
  - File: exercise 9
  - Kafka: exercise 10

- Sink
  - File: exercise 11
  - Kafka: exercise 12, exercise 13 (self study)

Structured Streaming: Cont..

- ETL
  - Apply all possible transformation as you do on normal dataframe: exercise 14

- Window aggregation
  - Analyze aggregated data over a period of time: exercise 15

Reference

- https://spark.apache.org/docs/latest/streaming-programming-guide.html
- https://spark.apache.org/docs/2.4.0/structured-streaming-kafka-integration.html
- https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html


Thank you!