Real Time Processing using Spark Streaming

Agenda

1. What is real time analytics (RTA) ?
2. Why we need it?
3. Use cases
4. Difference between Batch & Real time processing
5. Challenges in Real Time Processing
6. Intro to Spark Streaming
7. Streaming context
8. DStreams
9. Architecture
10. Window operations
11. Spark streaming context
12. Code example
13. Build real time pipeline using Spark Streaming
14. Fault tolerance & checkpoint

What is Real Time Analytics (RTA)?

- Processing large stream of data in real time which is generated continuously.
- 3 Vs of big data: Volume, Velocity, Variety
- Real time processing tasks include all the 3 Vs of big data paradigm.
- Streaming Data can be generated from thousands of data sources, which typically send in the data records simultaneously.
- Ability to collect and process TB's of streaming data to get insights

Why we need it?

- There are many big data technologies to handle huge volume of data in batch mode then why we need real time analytics?
- There are many real world applications where the response/action time to take any decision will be in seconds or microseconds.
- To derive insights from the huge dataset in microseconds we need Real Time Streaming capabilities.
- It is used for a wide variety of analytics including correlations, aggregations, filtering, and sampling.

Use Cases

1. Real Time Fraud Detection in Credit card & banking transactions. Credit card companies needs to crunch millions of transaction for identifying fraud
2. Digital marketing: Ad optimization & targeting based on each visit
3. Optimize and personalize content of Ad based on real time information.
4. Self Driving cars: Taking decision to stop, change direction, accelerate based on thousands of sensors in real time.
5. Social Media Trends:
    a. Twitter top trending keywords
    b. Facebook top trending topics
6. IOT sensors: To take any appropriate action many IOT sensors need real time information.
7. HFT Algorithmic Trading: This industry is based on real time analytics on all the stocks & financials data.

Use Cases

- Website traffic monitoring
- Network cluster monitoring
- Geo location targeting on real time based on user location.
- Real-time security intelligence operations to find threats
- Intelligence & Surveillance.
- Real-time video analytics to help with personalized, interactive experiences to the viewers
- E-commerce
- Real time in-store offers & recommendations.
- optimize user engagement based on user’s current behaviour.
- Pricing and analytics
- This list will go on, there are thousands of use case in each & every industry.

Big Companies using RTA

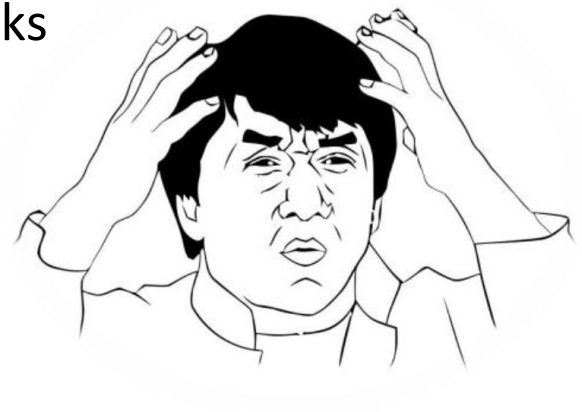
1. **Uber**: uses Spark Streaming in their continuous Streaming ETL pipeline to collect terabytes of event data every day from their mobile users for real-time analytics & pricing.
2. **Netflix**: uses Kafka and Spark Streaming to build a real-time online movie recommendation and data monitoring solution that processes billions of events received per day from different data sources.
3. **Pinterest**: uses Spark Streaming, MemSQL and Apache Kafka technologies to provide insight into how their users are engaging with Pins across the globe in real-time.
4. There are thousands of companies using RTA technologies.

Batch & Real Time systems

- Batch Processing:
  - It is where the processing happens of blocks of data that have already been stored over a period of time.
  - Batch Processing is performed majorly on the archival data to perform Big Data analytics and get useful insights of that data.
- Real Time Processing: It is processing and output of data as soon as input is received. There are 2 types of real time processing:
  - Near real time: When processing is not done on each & every single input in real time, but it is done on a window which is collection of few input events. It's like mini batch.
  - Pure real time: When processing is done on each & every input as soon as the input is received.

Batch & RTA Integration

- Many environments require processing same data in live streaming as well as batch post processing
- Existing framework cannot do both
  - Either do stream processing of 100s of MB/s with low latency
  - Or do batch processing of TBs / PBs of data with high latency
- Extremely painful to maintain two different stacks
  - Different programming models
  - Double the implementation effort
  - Double the number of bugs



Spark Streaming

- Extension of the core Spark API
- scalable, high-throughput, fault-tolerant stream processing of live data
- low latency processing
- Simple batch-like API for implementing complex algorithms
- It is for use cases which require a significant amount of data to be quickly processed as soon as it arrives.

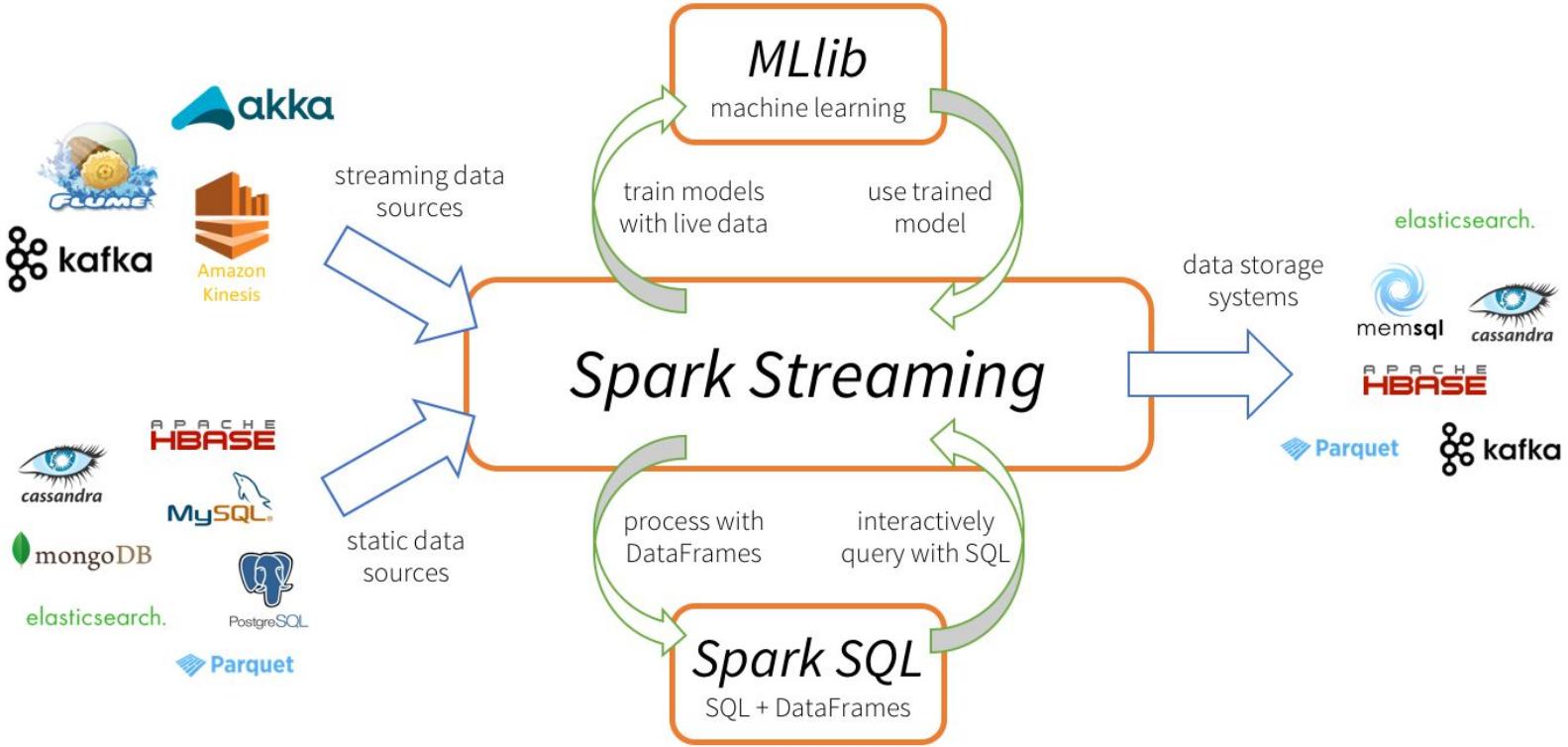
Input Output Connectors

1. Spark Streaming supports following input data sources (connectors):
    a. Kafka
    b. Flume
    c. HDFS/S3
    d. Kinesis
    e. Twitter
    f. TCP sockets
    g. Can define your own custom data source connector
2. Output can be persisted to following formats:
    a. HDFS
    b. Database
    c. Dashboard
    d. Cassandra
    e. HBase

Input Output Connectors

Spark streaming & core APIs

- Spark streaming can use all the core APIs of Spark, like:
  - Dataframes
  - RDDs & datasets
  - Spark sql
  - MLlib
  - GraphX
- Programming languages:
  - Scala
  - Java
  - Python


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

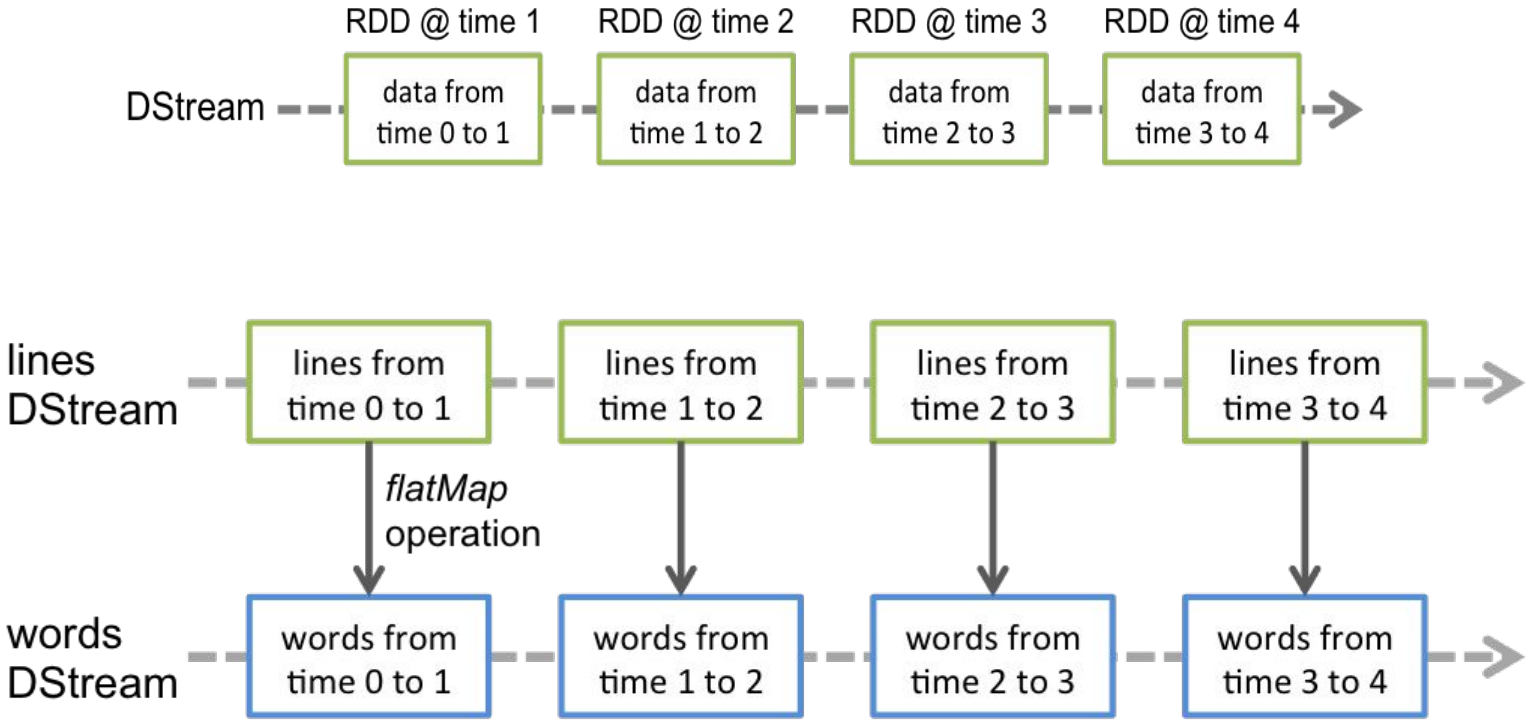
How Spark Streaming works?

- Spark Streaming divides a data stream into batches of X seconds called Dstreams.
- Discretized Stream is the basic abstraction provided by Spark Streaming. A sequence of RDDs.
- Spark Application processes the RDDs using Spark APIs, and the processed results of the RDD operations are returned in batches.

Mini Batch


input data stream
Spark Streaming
batches of input data
Spark Engine
batches of processed data

RDD level DStream



Window Operations

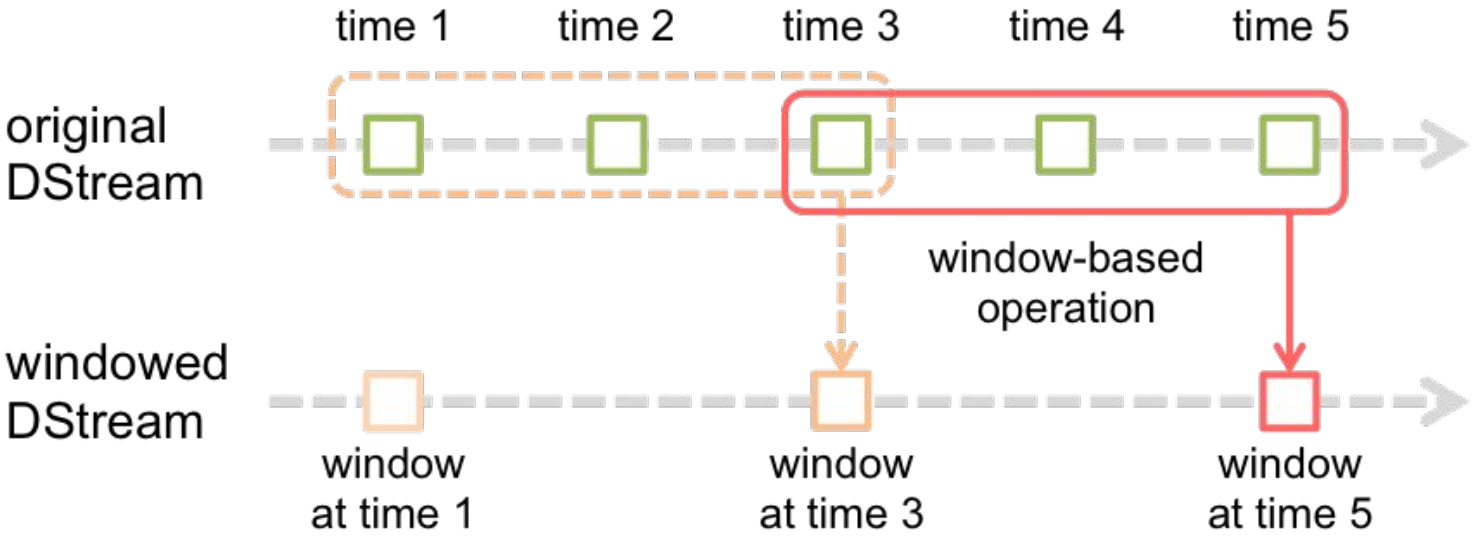
- Spark Streaming also provides *windowed computations*, which allow you to apply transformations over a sliding window of data
- every time the window *slides* over a source DStream, the source RDDs that fall within the window are combined and operated upon to produce the RDDs of the windowed Dstream
- *window length* - The duration of the window
- *sliding interval* - The interval at which the window operation is performed

ssc = StreamingContext(sc, *300, 10*)

Window length = 300 sec

Sliding interval = 10 sec

Window Operations



Competitors to spark stream...g

- Apache Storm: It operates on data in motion. The real time nature is due to its ability to operate on Event-at-a-time or Event stream processing (ESP). (pure real time)
- Spark streaming: It treats streaming computations as a series of deterministic batch computations on small time intervals. Micro-batched event processing (Near Real Time)
- Samza: It continuously computes results as data arrives which makes sub-second response times possible. Developed by Linkedin. Integration only with Kafka.
- IBM Infosphere Streams: IBM proprietary tool for streaming data processing. IBM specific programming language. Not open source.
- Apache Flink: A true stream processing framework. The use of algorithms in both streaming and batch modes.