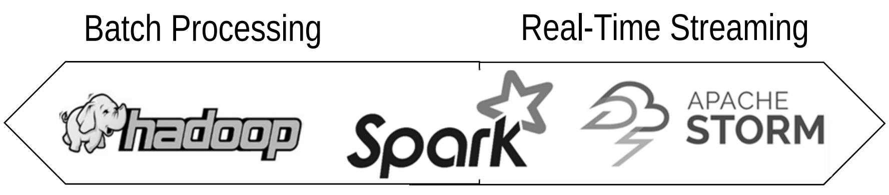
Big Data

Spark Streaming

Spark Streaming



Hadoop	Spark	Spark Streaming	Storm Trident	Storm
Batch Block	Batch RDD	Mini Batch RDD	TridentTuple TridentSpout TridentOperator	Tuple Spout Bolt

Spark Streaming

❖ Spark Streaming Characteristics

- Extension to the Spark Core API
- Live data streams can be processed
- Fault-tolerant and scalable
- High throughput (near) real-time data processing ➔ 0.5 s or longer
- Streaming data input from HDFS, Kafka, Flume, TCP sockets, Kinesis, etc.

Spark Streaming

❖ Spark Streaming Characteristics

- Stream processing high-level functions
  - Map, Reduce, Join, Window, etc.
- Processed Output saved on Filesystems, Databases, and Live Dashboards
- Spark ML (Machine Learning) functions and GraphX graph processing algorithms are fully applicable to streaming data

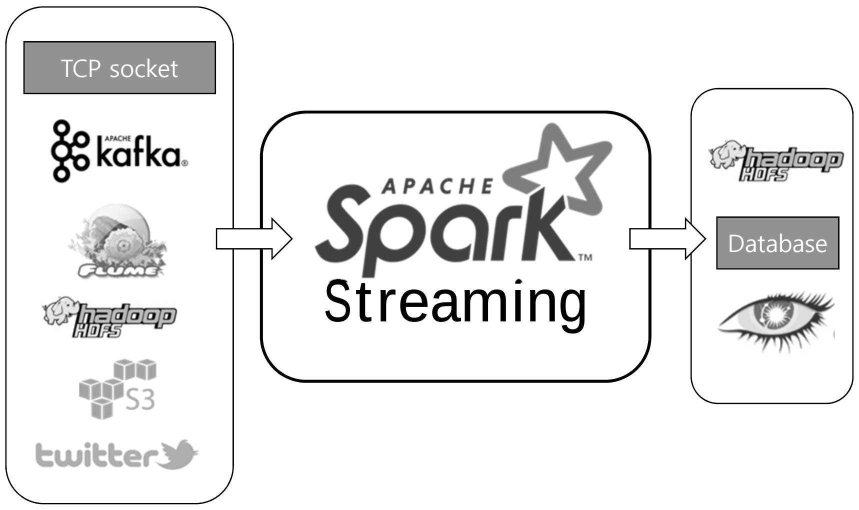
Spark Streaming

❖ **Spark Streaming Characteristics**

- **Spark uses (size controllable) micro-batch processing of data for real-time analysis**
  - **Hadoop uses batch processing of data, which is time consuming to obtain results**
- **Spark uses RDD to arrange data and recover from failures**

Spark Streaming

❖ **Spark Streaming Input & Output**



Spark Streaming

❖ **Streaming Receiver Types**

- **Basic**
  - **File systems**
  - **Socket connectors**
  - **Akka Actors**
  - **Sources directly available in Streaming Context API**

Spark Streaming

❖ **Streaming Receiver Types**

- **Custom**
  - **Requires implementing an user-defined receiver**
- **Advanced**
  - **Requires linking with systems that have extra dependencies**
  - **Kafka, Flume, Twitter**

Spark Streaming

❖ **Spark Streaming process**

1. Live input data stream received
2. Input data stream is divided into Mini-Batches called a DStream (Discretized Stream), which is saved as a small RDD every mini-batch period
3. Spark Stream engine cores process the mini-batches and generate a final output stream of mini-batches

Spark Streaming

❖ **DStream**

- DStream (Discretized Stream) is a continuous stream of data with high-level abstraction
- DStreams are created from input data stream sources (Kafka, Flume, Kinesis, etc.) or high-level processing operations on other DStreams

Spark Streaming

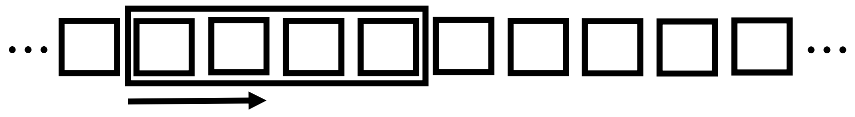
❖ DStream

- DStreams are represented as a sequence of small RDDs
- Mini-Batch size is 0.5 s or longer
- RDD is processed through the DAG
- Processing latency (through the DAG) has to be smaller than the mini-batch period

Spark Streaming

❖ Window Operations

- Window Length
  - Number of blocks (partitions) to conduct a RDD DAG process together



- Sliding Interval
  - Number of blocks (partitions) to slide the Window after a RDD process is conducted

Spark Streaming

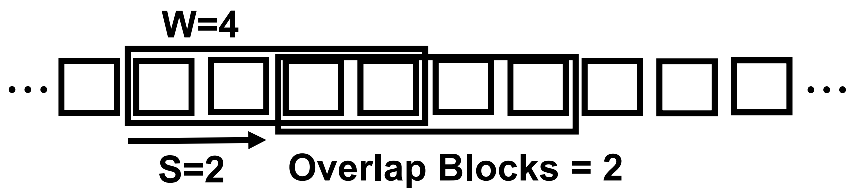
❖ **Transformations for Windows**

- Key parameters: windowLength, slideInterval
- **`window( )`**
- **`countByWindow( )`**
- **`reduceByWindow( )`**
- **`reduceByKeyAndWindow( )`**
- **`countByValueAndWindow( )`**
- etc.

Spark Streaming

❖ **Window Operations**

- **If Window Length > Sliding Interval, then**
  Overlap Blocks = (Window Length – Sliding Interval)
  will exist for each RDD process



- **Overlap Blocks help to analyze correlation (dependency) of sequential blocks of the streamed data**

Spark Streaming

❖ **Spark Streaming Examples**

- **IPTV or Web Page Live statistics**
  - **Channel or Page view of clicks**
  - **Use Kafka for buffering**
  - **Spark Streaming for processing**
  - **Draw a Heap Map of the current Channel or Page view clicks**

NETFLIX

Spark Streaming

❖ **Spark Streaming Examples**

- **Sales Product Type Monitoring**
  1. **Online Sales**
     - **Read through Kafka**
  2. **Department Store Sales**
     - **Read through Flume**
- **Join 2 live data streams into Spark Streaming**

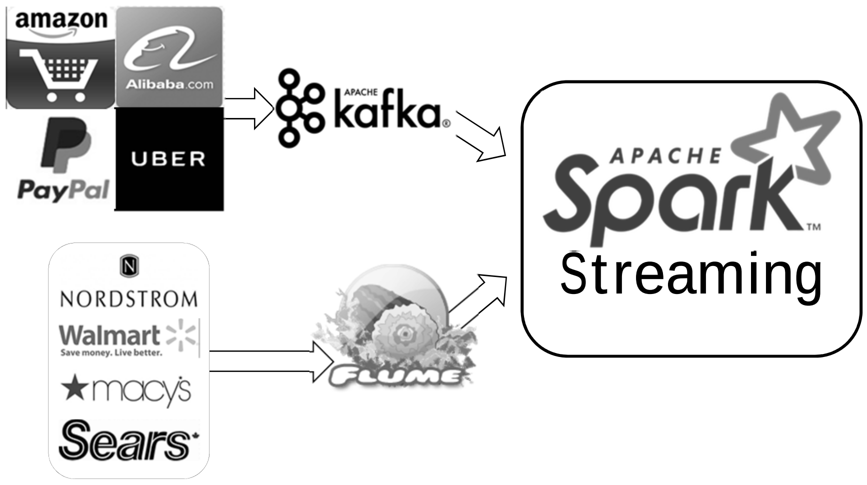
APACHE kafka

FLUME

APACHE Spark Streaming

Spark Streaming

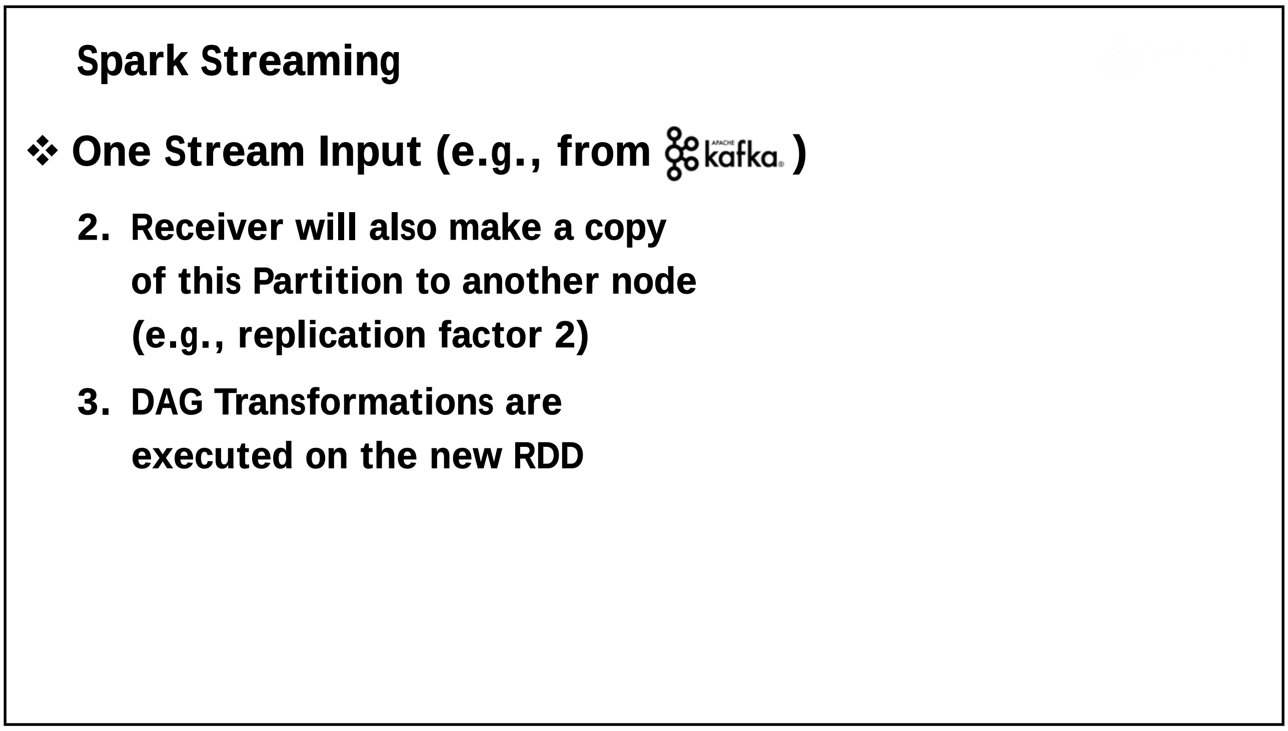
❖ **Streaming Receiver Types**



Spark Streaming

❖ **One Stream Input (e.g., from kafka)**

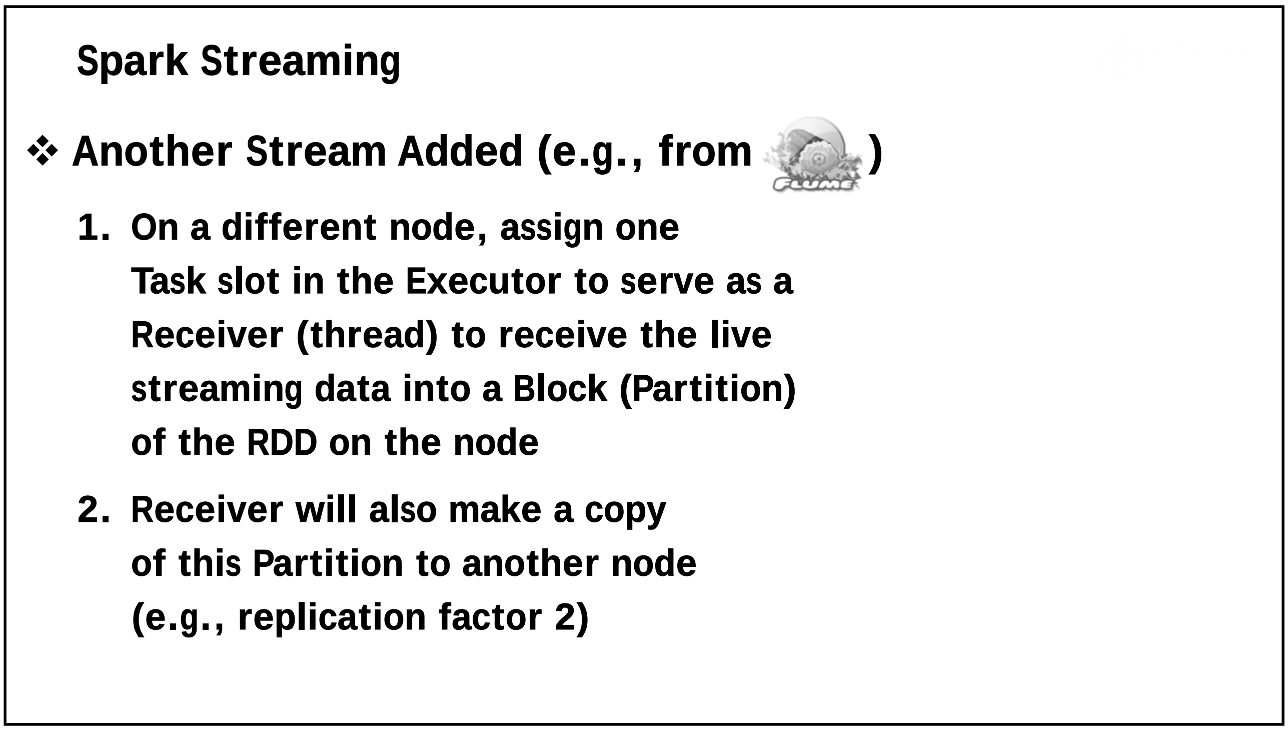
1. **One Task slot in the Executor will serve as a Receiver (thread) to receive the live streaming data into a Block (Partition) of the RDD on the node**



Spark Streaming

❖ One Stream Input (e.g., from kafka)

2. Receiver will also make a copy of this Partition to another node (e.g., replication factor 2)
3. DAG Transformations are executed on the new RDD



Spark Streaming

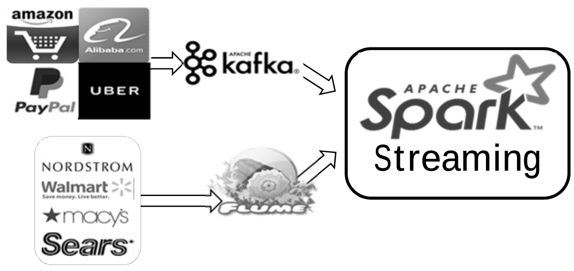
❖ Another Stream Added (e.g., from FLUME)

1. On a different node, assign one Task slot in the Executor to serve as a Receiver (thread) to receive the live streaming data into a Block (Partition) of the RDD on the node
2. Receiver will also make a copy of this Partition to another node (e.g., replication factor 2)

Spark Streaming

❖ **Another Stream Added (e.g., from)**

3. **DAG Transformations are executed on the new RDD**
4. **Union can be used to unify the two RDDs into one RDD**



Big Data

References

References

- Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*. 1st Edition. O'Reilly, 2015.
- Sameer Farooqui, Databricks, **Advanced Apache Spark Training**, Devops Advanced Class, Spark Summit East 2015, http://slideshare.net/databricks, www.linkedin.com/in/blueplastic, March 2015.
- Apache Spark documents (all documents and tutorials were used)
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html#working-with-key-value-pairs
  - https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html#rdd-persistence
- Wikipedia, www.wikipedia.org
- Stackoverflow, https://stackoverflow.com/questions
- Bernard Marr, "Spark Or Hadoop -- Which Is The Best Big Data Framework?," Forbes, Tech, June 22, 2015.
- Quick introduction to Apache Spark, https://www.youtube.com/watch?v=TgiBvKcGL24
- Wide vs Narrow Dependencies, https://github.com/rohgar/scala-spark-4/wiki/Wide-vs-Narrow-Dependencies

References

- Partitions and Partitioning, https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-rdd-partitions.html
- Neo4j, "From Relational to Neo4j," https://neo4j.com/developer/graph-db-vs-rdbms/ (last accessed Jan. 1, 2018).

Image Sources

- By Robivy64 at English Wikipedia [Public domain], via Wikimedia Commons
- Teravolt at English Wikipedia [CC BY 3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons
- By Konradr (Own work) [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons

Course Title
Big Data Emerging Technologies

❖ **Modules**

1. Big Data Rankings & Products
2. Big Data & Hadoop
3. Spark
4. Spark ML & Streaming
5. Storm
6. IBM SPSS Statistics Project