Welcome and Course Overview

Instructor's Welcome

Welcome to the Spark and Data Streaming course! You'll be first learning Apache Spark fundamentals which will help you build a Spark batch application, and then we'll be learning general concepts about streaming in the data world, and specifically learning to use Spark Structured Streaming. We'll prepare you to start using Structured Streaming APIs, and then dive into integrating Spark Structured streaming with Kafka, which you learned about in the previous course. By the end of the course you will complete a project that will prepare you to start building streaming applications at your job.

Course Outline

- Apache Spark fundamentals (RDD/DataFrame/Dataset)
- Actions/Transformations
- Spark Streaming/Structured Streaming
- Integration of Spark Streaming with Apache Kafka

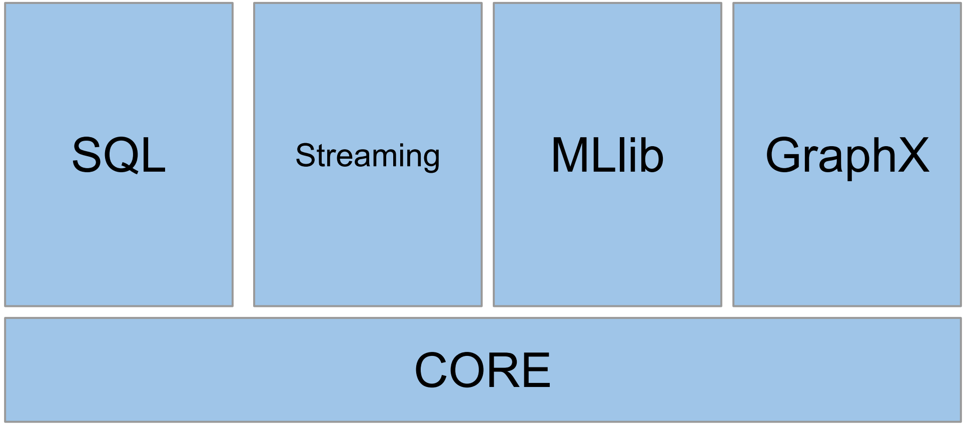
Instructor Comments on Job Skills in this Course

Introduce Spark Ecosystem

Introduce Spark Ecosystem Heading

- Explore Apache Spark components (focused on Core, SQL, and Streaming)
- RDD/DataFrame/Dataset
- Architecture of Apache Spark
- Action/Transformations

Spark Components



SQL

Streaming

MLlib

GraphX

CORE

Resilient Distributed Datasets (RDDs) in Spark

Spark RDD

- Resilient : Fault-tolerant
- Distributed : Data resides on multiple nodes
- Dataset : Records of the data

DEMO for RDD

DEMO for RDD

Partitioning in Spark

Two types of partitioning in Spark

- Hash (*partition = hash_code % number_of_partitions*)
- Range
  - tuples containing the same key will appear in the same machine

DataFrames

- DataFrames contain rows with Schema
- Common features of RDDs and DataFrames
    - Immutability, in-memory, resilience, distributed-computing capability
- Think of a DataFrame as a table in a relational database, or like a dataframe in the pandas library

Datasets

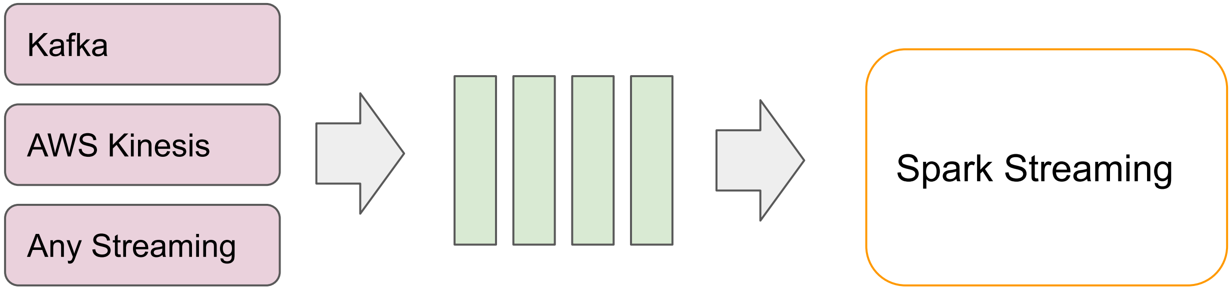
- Datasets are strongly typed
- A map to a relational schema
- Extension to DataFrame API
- Unfortunately not available for Python, so we won’t discuss them too much in this course

Introduce Spark Streaming/Structured Streaming

New Video

Discretized Stream

Architecture of D Stream



Kafka

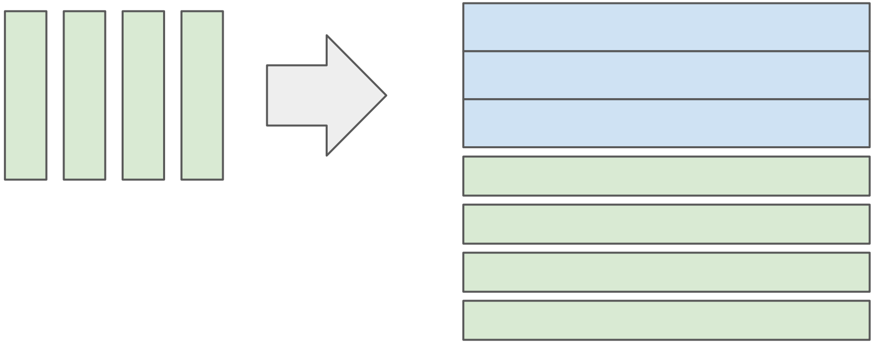
AWS Kinesis

Any Streaming

Spark Streaming

Structured Streaming

Structured Streaming



State Management in Structured Streaming

State Management in Spark Streaming (prior to v2.2.0)

- The state was persisted along with the checkpoint metadata
- Saving state-to-store was tightly coupled with Spark RDD tasks/jobs

State Management in Structured Streaming (v2.2.0~current)

- State management is now decoupled from metadata checkpointing
- Asynchronous to RDD execution
- Supports incremental state persistence

Structured Streaming Demo

Spark UI / DAGs / Phases

Spark UI Overview

- Directed Acyclic Graphs (DAGs) in Apache Spark
- Why DAGs are needed
- DAG Scheduler
- How to create DAGs
- How DAGs help in achieving fault tolerance
- How DAGs work in RDD
- Advantages of DAGs in Spark

Spark UI / DAGs

- DAG is an optimized execution plan with minimized data shuffling
- Spark creates a DAG when an action is called then submits it to the DAG Scheduler
- DAG Scheduler divides operators into stages of tasks
- Stages are passed onto Task Scheduler
- Primary node assigns the tasks to secondary nodes
- The lineage graph shows the history of RDD transformations

Spark UI / dags / stages Example

Spark Stages UI

Spark Stages

```
lines = spark.read.text(file_path).rdd.map(lambda x: x[0])

counts = lines.flatMap(lambda x: x.split(‘ ’)).map(lambda x: (x,
1)).reduceByKey()

output = counts.write.parquet(“file_path_to_save”)
```

Spark Stages

- ShuffleMapStage
  - An intermediate Spark stage in the physical execution of DAGs
  - Produces data for other stages
- ResultStage
  - Final stage in a Spark job
  - Result of an action

Spark Stages

```
lines = spark.read.text(file_path).rdd.map(lambda x: x[0])

counts = lines.flatMap(lambda x: x.split(‘ ’)).map(lambda x: (x,
1)).reduceByKey()

output = counts.write.parquet(“file_path_to_save”)
```

Spark UI / dags / stages+

Spark UI / dags / stages+ Walkthru

Spark StructType

Lesson Summary

So in this lesson, we learned the fundamentals of Spark’s core building blocks - RDDs, DataFrames, and Datasets - and how to use the Spark Web UI to monitor and debug Spark jobs.
You can now build a simple Spark application that focuses on batch processing.
Now in the next lesson you’re ready to dive into building Spark streaming applications using structured streaming APIs.