Spark Training

[jacek@japila.pl](mailto:jacek@japila.pl)

GOAL

Increase the success rate of all the available Spark certifications available out there such as Databricks, Hortonworks, MapR, Cloudera

**AGENDA**

Day 1 – The essentials of Spark SQL

Day 2 – Advanced Spark SQL

Day 3 – Spark Structured Streaming

Day 4 – Spark MLlib

Day 5 – Advanced Apache Spark (Core)

Get started

First exercise

Always use SparkSession (:type spark)

Always avoid SparkContext (:type sc)

The essentials of Spark SQL

* Structured queries describe using SQL and Dataset (and DF)
* Good ol' SQL statements
* Dataset(and Frame) data abstractions -> distributed computations
* Encoder for storage and performance optimizations -> Reducing garbage collection

SparkSession

1. Entry point to Spark SQL and Spark in general these days
2. Use SparkSession.builder Fluent API to create one
3. Allows creating local Datasets -> spark.range(numberOfRecords)
4. Loading datasets using load
5. Spark-shell gives you one instance as spark
6. Switch to Mastering SparkSQL -> DataSource API Managing datasets in external data sources

DataSource API – Reading and Writing Datasets

1. Loading dataets using SparkSession.read
2. Writing Datasets using Dataset.write -> Dataset = a distributed computation
3. Loading and writing operators create source and sink nodes in a data flow graph
4. Switch to Mastering Spark SQL

Reading/Loading Datasets

val dataset = spark.read.format("json").save("dailies")

1. DataFrame.write
   1. format
   2. mode
   3. option and options
   4. partitioBy, bucketBy, sortBy
   5. insertInto, save, saveAsTable

Ad-hoc local DataSets

1. Seq(…).toDF("col1","col2",…) for local DataFrames
2. Seq(…).toDS for local Datasets
3. Use import spark.implicits.\_
4. All Scala Collections supported (almost)
5. Switch to Mastering Spark SQL

Schema

1. Schema = StructType with one or many StructFields
2. Implicit (inferred) or explicit
3. Dataset.printSchema
4. Schema is your case class(es)

Dataset Columns

1. Column is a function that generates a value per row

Column Operators

1. Special star column reference
2. Operators to create compound columns
   1. As alias
   2. === for equality
   3. Desc, desc\_nulls\_first, desc\_nulls\_last (and for asc)
   4. getItem
   5. over
   6. cast
   7. when and otherwise
3. Read up on Column's scaladoc

Exercice using flatmap

1. Create a dataset with a column of type array
2. Use flatmap to expand the array column
3. Compare performance of flatMap and explode

Aggregation Functions

Aggregate functions accept a group of records as input unlike regular functions that act on a single record.

Available among standard functions

AGG

1. Agg applies an aggregate function to records in Dataset
2. Entire Dataset acts as a single group. groupBy used to define groups.
3. Creates DataFrame, hence considered untyped due to Row inside
4. Typed variant available
5. Switch to Mastering Spark SQL
6. Develop a standalone SparkSQL app, use max function with agg or select perators.

use GroupBy operator and max function to find the highest sales, use join to find names of bestsellers