

Big data: architectures and data analytics

Spark SQL

UDFs: User Defined Functions

- Spark SQL provides a set of system predefined functions
 - `hour(Timestamp)`, `abs(Integer)`, ..
 - Those functions can be used in some transformations (e.g., `selectExpr(..)`, `sort(..)`) but also in the SQL queries
- Users can define their personalized functions
 - They are called User Defined Functions (UDFs)

3

UDFs: User Defined Functions

- UDFs are defined/registered by invoking the `udf().register(String name, UDF function, DataType datatype)` on the `JavaSparkSession`
 - `name`: name of the defined UDF
 - `function`: lambda function/class used to specify how the parameters of the function are used to generate the returned value
 - One or more input parameters
 - One single returned value
 - `datatype`: SQL data type of the returned value

4

UDFs: User Defined Functions – Example

- Define a UDFs that, given a string, returns the length of the string


```
// Create a Spark Session
SparkSession ss = SparkSession.builder().appName("Spark
Example").getOrCreate();
// Define the UDF
// name: length
// input: String
// output: Integer
ss.udf().register("length", (String name) -> name.length(),
DataTypes.IntegerType);
```

5

UDFs: User Defined Functions – Example

- Use of the defined UDF in a selectExpr transformation


```
// Create a Spark Session
Dataset<Row> result=
inputDF.selectExpr("length(name) as size");
```
- Use of the defined UDF in a SQL query


```
// Create a Spark Session
Dataset<Row> result=
ss.sql("SELECT length(name) FROM profiles");
```

6

UDAFs: User Defined Aggregate Functions

- Sparks allows defining personalized aggregate function
 - They are used to aggregate the values of a set of tuples
- They are based on the implementation of the [org.apache.spark.sql.expressions.UserDefinedAggregateFunction](#) abstract class

7

UDAFs: User Defined Aggregate Functions

- The definition of the class associated with an aggregate function is associated with many variables and methods
 - Definition of input, intermediate, and returned schemas
 - Definition of the update and merge procedures
 - Update the internal buffer value by combining it with a new input record
 - Merge the local results of two partitions
 - Convert the internal buffer into the final returned result

8