SPARK SQL

Sundharakumar KB

Department of Computer Science and Engineering
School of Engineering

Shiv Nadar University Chennai

SHIV NADAR UNIVERSITY CHENNAI

Spark SQL

- Dataframes instead of RDDs

- Extends RDDs to dataframe objects
- DF :
  - Contain row elements
  - Can run SQL queries
  - Can have a schema
  - Read and write to JSON, HIVE, csv, etc.
  - Communicates with JDBC/ODBC, tableau, etc.

Spark SQL

- Instead of creating sparkcontext, we must create sparksession to use it with spark sql.
- Eg: from pyspark.sql import SparkSession, Row
- Sparksession is the entry point to use dataframes.
- To use SparkSession, you must use SparkSession.builder

Spark SQL

- inputData = spark.read.json(data)
- inputData.createOrReplaceTempView(“myView”)
- resultDF = spark.sql(“select foo from xyz ORDER BY foobar”)

Spark SQL

- resultDF.show()
- resultDF.select(“someFieldName”)
- resultDF.filter(resultDF(“someFieldName”>200))
- resultDF.groupby(resultDF(“someFieldName”)).mean()
- resultDF.rdd().map(mapperFunction)

- Most of the current spark operations deals with the Dataframes more than RDDs because of the flexibility that dataframes offer.

Spark SQL

- Spark SQL exposes JDBC/ODBC server.
- Can also be connected using the spark shell.
- Can also be used with hive using hiveCtx.cacheTable("tablename").
- Provides SQL shell to directly create new tables or query from existing tables.

User Defined Functions

- From pyspark.sql.types import IntegerType

- Def square(x):
- return x*x

- Spark.udf.register(“square”, square, IntegerType())

- Df = spark.sql(“select square(“SomeNumField”) from tableName”)