Lab 5.1: Exploring Grouping

Overview

In this lab, we will work with the Github archive that contains activity for a single day. We'll analyze it to find the top contributors for that day.

Builds on

None - but you must have the spark shell running.

Run time

25-35 minutes

Github Activity Archive

You've already worked with the data file (***spark-labs/data/github.json***) that contains a log of all github activity for one day. In previous labs you should already have

- Loaded the data, and viewed the schema.
- Selected the actor column, which has a nested structure, and worked with some of its subelements.
- We illustrate doing that below.

```
// Scala
// Load the file into Spark
> val githubDF=spark.read.json("spark-labs/data/github.json")
// Select the actor column
> val actorDF = githubDF.select('actor)
// Print actor schema
> actorDF.printSchema
// Select the actor's login value - note how we
// Use a SQL expression in the select, not a Column
> actorDF.select("actor.login").limit(5).show
```

```
# Python
# Load the file into Spark
> githubDF=spark.read.json("spark-labs/data/github.json")
# Select the actor column
> actorDF = githubDF.select(githubDF.actor)
# Print actor schema
> actorDF.printSchema()
# Select the actor's login value - note how we
# Here, we're using a SQL expression in the select, not a Column (it's simpler)
> actorDF.select("actor.login").limit(5).show()
```

Tasks

- If you haven't already done the steps above, then do so now.
    - You'll only reuse the githubDF dataframe in this lab.
    - The other statements are to practice working with the schema, which is complex.

Query the Data by Actor's Login Value

Tasks

- Query the github data for how many entries exist in the data for each actor's login. Use the DSL.
    - **Hints**:
        - You'll want to group the data by the actor's login.
        - You'll probably want to use an SQL expression to express the actor's login, not a column value.
        - You want a count of entries for each login value.
- Show a few rows of this data.
- Lastly, find the 20 logins with the largest number of contributions, and display them.

[Optional] Use SQL

Tasks

- Optionally, try doing the above query in SQL.
  - It's pretty much standard SQL, so if you know that well, it's not very complex.
  - Remember to create a temporary view (`createOrReplaceTempView`)

Summary

The task we did in this lab is not overly complex, but it's also not trivial. Spark SQL makes it reasonable for us to accomplish this in a short lab, either using the DSL, or using SQL.

If you wanted to do this using RDDs, it would be a much more complex series of transformations - starting with the messy ones of parsing the JSON data. This is why Spark SQL is becoming the standard API for working with Spark.