Module 6: Dataframes and Spark SQL

Case Study II Solution

edureka!

Case Study: Mobile App Store
Domain: Telecom

A telecom software provider is building an application to monitor different telecom components in the production environment. For monitoring purpose, the application relies on log files by parsing the log files and looking for potential warning or exceptions in the logs and reporting them.

The Dataset contains the log files from different components used in the overall telecom application.

Tasks:

The volume of data is quite large. As part of the R&D team, you are building a solution on spark to load and parse the multiple log files and then arranging the error and warning by the timestamp.

1. Load data into Spark DataFrame
2. Find out how many 404 HTTP codes are in access logs.
3. Find out which URLs are broken.
4. Verify there are no null columns in the original dataset.
5. Replace null values with constants such as 0
6. Parse timestamp to readable date.
7. Describe which HTTP status values appear in data and how many.
8. How many unique hosts are there in the entire log and their average request
9. Create a spark-submit application for the same and print the findings in the log

Solution:

- You should enter inside the PySpark Shell. ***Type*** ***pyspark2*** to enter the python spark shell for running further codes.

Step 1: Load data into Spark DataFrame

```
raw = spark.read.text(“use_cases/access.clean.log”)
```

```
df = raw.rdd.map(lambda r:r[0].split(" ")).map(lambda
arr:Row(remote_host=arr[0], timestamp=arr[3].replace("[",""),
request_type=arr[5],url=arr[6],status_code=arr[8])).toDF()
```

```
df.registerTempTable("logs")
```

Step 2: Find out how many 404 HTTP codes are in access logs

```
spark.sql("select count(*) from logs where status_code='404'").show()
```

Step 3: Find out which URLs are broken

```
spark.sql("select count(*) from logs where status_code=‘204'").show()
```

Step 4: Verify there are no null columns in the original dataset

```
filterCond = " is Null or ".join(df.columns) + " is Null”
spark.sql("select count(*) from logs where " + filterCond).show()
```

Step 5: Replace null values with constants such as 0

```
df1 = spark.sql("select remote_host, timestamp, request_type, url,
nvl(cast(status_code as String),'404') as status_code from logs”)
```

Step 6: Parse timestamp to readable date

spark.sql("select remote_host, timestamp, request_type, url, status_code, to_date(cast(unix_timestamp(timestamp, 'dd/MMM/yyyy') as timestamp)) as date from logs").show()

Step 7: Describe which HTTP status values appear in data and how many

spark.sql("select status_code, count(*) as cnt from logs group by status_code order by cnt desc").show()

Step 8: How many unique hosts are there in entire log and their average request

spark.sql("select remote_host, count(*) from logs group by remote_host").show()

Step 9: Create a spark-submit application for the same and print the findings