Mattia Danese
CS 119 - Big Data
Professor Singh
Quiz 4: Log File Analysis using Spark RDDs

Question 1

The percentage of each type of response in this log are:

Information responses: 0.00%
Successful responses : 14.26%
Redirection responses : 28.56%
Client Error responses: 28.56%
Server Error responses: 28.63%

Commands I ran:

- I used the PySpark library and defined the script below
- `python3 script.py`

Question 2

The top 5 IP addresses that generate the most client errors are:

122.155.216.51 → 2 client errors
40.237.64.134 → 2 client errors
252.215.192.215 → 2 client errors
175.115.37.123 → 2 client errors
204.12.91.26 → 2 client errors

Commands I ran:

- I used the PySpark library and defined the script below
- `python3 script.py`

Script.py

```
from pyspark.sql.functions import col
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("miniProject").setMaster("local[*]")
sc = SparkContext.getOrCreate(conf)
spark = SparkSession(sc)

original_RDD = spark.read.text("logfiles.log").rdd
ip_and_status_RDD = original_RDD.map(lambda row: (row.value.split(" ")[0],
row.value.split(" ")[8]))
```

```
total_logs = float(ip_and_status_RDD.count())

info_repsonses_RDD  = ip_and_status_RDD.filter(lambda row: row[1][0] == '1')
succ_repsonses_RDD  = ip_and_status_RDD.filter(lambda row: row[1][0] == '2')
redir_repsonses_RDD = ip_and_status_RDD.filter(lambda row: row[1][0] == '3')
cerr_repsonses_RDD  = ip_and_status_RDD.filter(lambda row: row[1][0] == '4')
serr_repsonses_RDD  = ip_and_status_RDD.filter(lambda row: row[1][0] == '5')

print("Percent of Information responses :
{}%".format(round((float(info_repsonses_RDD.count()) / total_logs) * 100.0,
2)))
print("Percent of Successful responses  :
{}%".format(round((float(succ_repsonses_RDD.count()) / total_logs) * 100.0,
2)))
print("Percent of Redirection responses :
{}%".format(round((float(redir_repsonses_RDD.count()) / total_logs) * 100.0,
2)))
print("Percent of Client Error responses:
{}%".format(round((float(cerr_repsonses_RDD.count()) / total_logs) * 100.0,
2)))
print("Percent of Server Error responses:
{}%".format(round((float(serr_repsonses_RDD.count()) / total_logs) * 100.0,
2)))
print()

columns = ["ip", "status code"]
cerr_repsonses_DF = cerr_repsonses_RDD.toDF()
cerr_repsonses_DF2 = cerr_repsonses_DF.toDF(*columns)
cerr_repsonses_DF3 = cerr_repsonses_DF2.groupBy("ip").count()
cerr_repsonses_DF4 = cerr_repsonses_DF3.sort(col("count").desc())

print("The top 5 IP addresses that generate the most client errors are:")
for ip, count in cerr_repsonses_DF4.take(5):
   print("{}{} -> {} client error{}".format(ip, "".join([" " for x in
range(15-len(ip))]),count,"" if count == 1 else "s" ))
```