Semantix - Desafio Engenheiro de Dados

August 8, 2018

1 Semantix

1.1 Desafio Engenheiro de Dados

Thiago de Lucena Nascimento
Email: thiago_lucena@hotmail.com.br
LinkedIn: https://br.linkedin.com/pub/thiago-de-lucena-nascimento/85/b80/587
Contato: (11)96358-7940

1.1.1 Qual o objetivo do comando cache em Spark?

Possibilita a persistencia dos dados em memória, permitindo que sejam reutilizados em etapas seguintes evitando um novo processamento.

1.1.2 O mesmo código implementado em Spark é normalmente mais rápido que a implementação equivalente em MapReduce. Por quê?

O MapReduce utiliza-se do disco (HDFS) para realizar a gravação dos resultados intermediários em uma atividade de processamento, ao passo que o Spark utiliza-se da memória. O processamento em memória é até 100x mais rápido.

1.1.3 Qual é a função do SparkContext?

Estabelecer a conexão com o ambiente de execução do Spark, permitindo acesso à todas as suas funcionalidades.

1.1.4 Explique com suas palavras o que é Resilient Distributed Datasets (RDD).

É uma coleção de elementos de dados particionados distribuída e imutável.

1.1.5 GroupByKey é menos eficiente que reduceByKey em grandes dataset. Por quê?

O GroupByKey realiza o agrupamento de todos os dados de diferentes partições para só então realizar a operação. O reduceByKey realiza o agrupamento dos dados por partição para depois realizar a operação. Isso implica em um volume menor de dados trafegando na rede.

1.1.6 Explique o que o código Scala abaixo faz.

val textFile = sc.textFile("hdfs://...") #CRIA UM RDD A PARTIR DE UM ARQUIVO DO HDFS val counts = textFile.flatMap(line => line.split(" ") #APLICA O SPLIT PARA QUEBRAR O RDD EM PALAVRAS .map(word => (word, 1)) #FAZ UM MAPEAMENTO "CHAVE, VALOR" DE CADA PALAVRA ATRIBUINDO O VALOR 1 .reduceByKey(_ + _) #FAZ REDUÇÃO SUMARIZANDO OS VALORES POR PALAVRAS, RESULTANDO NA QUANTIDADE TOTAL DE CADA PALAVRA counts.saveAsTextFile("hdfs://...") #SALVA O ARQUIVO COM A CONTAGEM DE REPETIÇÃO DAS PALAVRAS NO HDFS

2 HTTP requests to the NASA Kennedy Space Center WWW server

```
In [1]: # Importando bibliotecas
        from pyspark.sql import SparkSession
        from pyspark.sql import SQLContext
        from pyspark.sql.functions import regexp_extract
```

```
In [2]: # Criando o Spark Session
        spSession = SparkSession.builder.master("local").appName("Semantix-Desafio").config("sp
```

```
In [3]: # Criando o SQL Context
        sqlContext = SQLContext(sc)
```

```
In [4]: # Unindo os dois arquivos
        Jul95 = sqlContext.read.text("access_log_Jul95")
        Aug95 = sqlContext.read.text("access_log_Aug95")
        NASA_Jul95_Aug95 = Jul95.union(Aug95)
```

```
In [5]: # Separando as colunas através de expressões regulares
        NASA = NASA_Jul95_Aug95.select(regexp_extract('value', r'^([^\s]+)', 1).alias('HOST'),
        regexp_extract('value', r'^.*\[(\d\d/\w{3}/\d{4}:\d{2}:\d{2}:\d{2} -\d{4})]', 1).alias
        regexp_extract('value', r'^.*"\w+\s+([^\s]+)\s+HTTP.*"', 1).alias('REQUISICAO'),
        regexp_extract('value', r'^.*"\s+([^\s]+)', 1).cast('integer').alias('RETORNO'),
        regexp_extract('value', r'^.*\s+(\d+)$', 1).cast('integer').alias('BYTES'))
```

```
In [6]: # Criando uma Tabela Temporaria
        NASA.createOrReplaceTempView("NASA_TEMP_TAB")
```

2.1 Questões

2.1.1 1. Número de hosts únicos.

Nesta questão fiquei em dúvida de como deveriam ser apresentados os resultados. Então, fiz duas consultas, uma com o total e outra listando os hosts.

```
In [7]: spSession.sql("SELECT SUM(CONTAGEM) as TOTAL \
                      FROM (SELECT COUNT(1) as CONTAGEM \
                            FROM NASA_TEMP_TAB \
                            GROUP BY HOST \
                            HAVING COUNT(1) <2) \
                      DISTINTOS").show()
```

```
+-----+
|TOTAL|
+-----+
| 9269|
+-----+
```

```
In [8]: spSession.sql("SELECT COUNT(1) as CONTAGEM, \
                              HOST \
                              FROM NASA_TEMP_TAB \
                              GROUP BY HOST \
                              HAVING COUNT(1) <2").show()
```

```
+--------+--------------------+
|CONTAGEM|                HOST|
+--------+--------------------+
|       1|     193.166.184.116|
|       1|      204.120.34.242|
|       1|dutton.cmdl.noaa.gov|
|       1|    rickf.seanet.com|
|       1|  vdfcomm.vdfnet.com|
|       1|ldvgpi33.ldv.e-te...|
|       1|tenebris.rutgers.edu|
|       1|       144.191.11.42|
|       1|inf-pc43.fbm.htw-...|
|       1|       conan.ids.net|
|       1|   obiwan.tdtech.com|
|       1|      204.180.143.17|
|       1|n1-28-222.macip.d...|
|       1|      chi007.wwa.com|
|       1|       137.148.36.27|
|       1|       129.219.88.17|
|       1|       nu.sim.es.com|
|       1|jobstgb1.bradley.edu|
|       1|ip-pdx1-51.telepo...|
|       1|       194.20.140.83|
+--------+--------------------+
only showing top 20 rows
```

2.1.2 2. O total de erros 404.

```
In [9]: spSession.sql("SELECT COUNT(1) as TOTAL_ERRO \
                       FROM NASA_TEMP_TAB \
                       WHERE RETORNO=404").show()
```

```
+----------+
|TOTAL_ERRO|
+----------+
|     20901|
+----------+
```

2.1.3 3. Os 5 URLs que mais causaram erro 404.

```
In [10]: spSession.sql("SELECT COUNT(1) as TOTAL, \
                        REQUISICAO as URL \
                        FROM NASA_TEMP_TAB \
                        WHERE RETORNO=404 \
                        GROUP BY REQUISICAO \
                        ORDER BY COUNT(1) DESC \
                        LIMIT 5").show()
```

```
+-----+--------------------+
|TOTAL|                 URL|
+-----+--------------------+
| 2004|/pub/winvn/readme...|
| 1732|/pub/winvn/releas...|
|  682|/shuttle/missions...|
|  426|/shuttle/missions...|
|  384|/history/apollo/a...|
+-----+--------------------+
```

2.1.4 4. Quantidade de erros 404 por dia.

```
In [11]: spSession.sql("SELECT COUNT(1) as TOTAL_ERRO, \
                        SUBSTR(TIMESTAMP, 1, 11) as DIA \
                        FROM NASA_TEMP_TAB \
                        WHERE RETORNO=404 \
                        GROUP BY SUBSTR(TIMESTAMP, 1, 11)").show()
```

```
+----------+-----------+
|TOTAL_ERRO|        DIA|
+----------+-----------+
|       291|02/Jul/1995|
|       305|21/Aug/1995|
|       373|06/Aug/1995|
|       257|16/Jul/1995|
|       537|07/Aug/1995|
|       263|11/Aug/1995|
|       336|27/Jul/1995|
|       570|07/Jul/1995|
```

```
|       406|17/Jul/1995|
|       254|15/Jul/1995|
|       465|18/Jul/1995|
|       336|26/Jul/1995|
|       304|03/Aug/1995|
|       256|18/Aug/1995|
|       271|17/Aug/1995|
|       287|14/Aug/1995|
|       398|10/Jul/1995|
|       359|04/Jul/1995|
|       312|20/Aug/1995|
|       428|20/Jul/1995|
+----------+-----------+
only showing top 20 rows
```

2.1.5 5. O total de bytes retornados.

```
In [12]: spSession.sql("SELECT SUM(BYTES) as TOTAL_BYTES \
                        FROM NASA_TEMP_TAB").show()
```

```
+-----------+
|TOTAL_BYTES|
+-----------+
|65524314915|
+-----------+
```