

Desafio_Spark

February 16, 2020

0.0.1 Processamento de Arquivos de LOGs com Spark - 02/2020

0.0.2 MARCIO DE LIMA

0.0.3 DADOS Fornecidos

Fonte oficial do dataset: <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>

Os dois conjuntos de dados possuem todas as requisições HTTP para o servidor da NASA Kennedy Space Center WWW na Flórida para um período específico.

0.0.4 Colunas

Arquivos em ASCII com as colunas:

Host , um hostname quando possível, caso contrário o endereço de internet se o nome não puder ser identificado.

Timestamp no formato "DIA/MÊS/ANO:HH:MM:SS TIMEZONE"

Requisição (entre aspas)

Código do retorno HTTP

Total de bytes retornados

0.1 ***** Atenção: *****

Ambiente de Desenvolvimento desse fonte: Linux, Java JDK 8 , Apache Spark 2.4.3 e Python 3.7 com Jupyter LAB.

```
In [1]: #Importando as bibliotecas
```

```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SQLContext
from pyspark.sql import Row
```

```
In [2]: # Define o Spark Context, pois o job será executado via linha de comando com o spark-s
```

```
conf = SparkConf().setMaster("local").setAppName("DesafioSpark")
sc = SparkContext(conf = conf)
```

```
In [3]: # Criando o SQL Context para trabalhar com Spark SQL
```

```
sqlContext = SQLContext(sc)
```

```
In [4]: #Versão do Spark utilizada
```

```
print(sc.version)
```

2.4.3

```
In [5]: # Criando os RDDs a partir dos arquivos fornecidos, encoding => iso-8859-1
logJulRDD = sc.textFile("dados/NASA_access_log_Jul95", use_unicode=False).map(lambda x:
logAgoRDD = sc.textFile("dados/NASA_access_log_Aug95", use_unicode=False).map(lambda x:

In [6]: #Juntando os dois RDDs em 1 único RDD
joinedLinhas = logJulRDD.union(logAgoRDD)

In [7]: #Contagem de Linhas de LOG
joinedLinhas.count()

Out[7]: 3461613

In [8]: # O Join insere 1 linha em branco no final do arquivo, alem disso, o arquivo pode cont
joinedLinhasL = joinedLinhas.filter(lambda linha: linha != '')

In [9]: joinedLinhasL.take(10)

Out[9]: ['199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 62
'unicom6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/
'199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission
'burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.l
'199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73
'burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif I
'burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/li
'205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.h
'd104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3
'129.94.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074']

In [10]: # Limpando a memória
del joinedLinhas, logJulRDD, logAgoRDD

In [11]: #Persistindo na memória
joinedLinhasL.persist()
joinedLinhasL.cache()

Out[11]: PythonRDD[9] at RDD at PythonRDD.scala:53

In [12]: #Função de limpeza e tratamento dos dados, linha a linha
def prepararDados(linha):

    attList = linha.split('\n')
    attList_1 = attList[0].split(" ")

    try:
        data = attList_1[3].strip()[1:7] if attList_1[3] != '' else ''
        erro_data = 0
    except:
```

```

        data = '01/Jan'
        erro_data = 1
    finally: {}

    try:
        attList_2 = linha.split('/1.0\')[1].split(" ")
        httpCode = attList_2[1].strip() if attList_2[1] != '-' and attList_2[1] != '' else ''
        erro_httpCode = 0
    except:
        httpCode = '0'
        erro_httpCode = 1
    finally: {}

    try:
        attList_2 = linha.split('/1.0\')[1].split(" ")
        bytesTransf = attList_2[2].strip() if attList_2[2] != '-' else '0'
        erro_bytes = 0
    except:
        bytesTransf = '0'
        erro_bytes = 1
    finally: {}

    host = attList_1[0].strip() if attList_1[0] != '' else ''

    try:
        url = attList[1].replace("GET ", "").replace(" HTTP/1.0", "").strip()
        erro_url = 0
    except:
        url = ''
        erro_url = 1
    finally: {}

    valores = Row(Host = host, Data = data, HttpCode = int(httpCode), Bytes = int(bytesTransf))
    return valores

```

```

In [13]: # Aplicando a função em todo o dataSet
         joinedLinhasLimpo = joinedLinhasL.map(prepararDados)

```

```

In [14]: #Persistindo na memória
         joinedLinhasLimpo.persist()
         joinedLinhasLimpo.cache()

```

```

Out[14]: PythonRDD[10] at RDD at PythonRDD.scala:53

```

```

In [15]: # Criando DataFrame
         df = sqlContext.createDataFrame(joinedLinhasLimpo)

```

```

In [16]: # Criando DataFrame
         df = sqlContext.createDataFrame(joinedLinhasLimpo)

```

```
# Zerando valores NA , caso existam
data_df = df.fillna(0)
```

```
In [17]: # Mostrando e colocando no cache os dados
data_df.persist()
data_df.cache()
```

```
# Limpando a memória
del joinedLinhasLimpo, df

data_df.show()
```

Bytes	Data	ErroConversaoBytes	ErroConversaoData	ErroConversaoHttp	ErroConversaoUrl
6245	01/Jul	0	0	0	0
3985	01/Jul	0	0	0	unicomp6
4085	01/Jul	0	0	0	199
0	01/Jul	0	0	0	burger
4179	01/Jul	0	0	0	199
0	01/Jul	0	0	0	burger
0	01/Jul	0	0	0	burger
3985	01/Jul	0	0	0	205
3985	01/Jul	0	0	0	
7074	01/Jul	0	0	0	123
40310	01/Jul	0	0	0	unicomp6
786	01/Jul	0	0	0	unicomp6
1204	01/Jul	0	0	0	unicomp6
40310	01/Jul	0	0	0	
786	01/Jul	0	0	0	
1204	01/Jul	0	0	0	
0	01/Jul	0	0	0	123
1713	01/Jul	0	0	0	199
3977	01/Jul	0	0	0	ppptky39
34029	01/Jul	0	0	0	net-1-

only showing top 20 rows

```
In [18]: # Registrando o dataframe como uma Temp Table para a execução dos SQLs
data_df.createOrReplaceTempView("linhasTB")
```

0.1.1 Questões

0.1.2 1) Número de hosts únicos

```
In [19]: joinedLinhasL.map(lambda linha: linha.split(" ")[0]).distinct().count()
```

```
Out[19]: 137979
```

0.1.3 Resposta: 137979 hosts

0.1.4 2) O total de erros 404

```
In [20]: sqlContext.sql("select count(Host) as resultado from linhasTB where HttpStatusCode = '404'")
```

```
+-----+
|resultado|
+-----+
|    20698|
+-----+
```

0.1.5 Resposta: 20698 erros

0.1.6 3) Os 5 URLs que mais causaram erro 404

```
In [21]: # Executando SQL - Top 5
```

```
consulta = sqlContext.sql("select Url as url , count(HttpStatusCode) as resultado from linhasTB where HttpStatusCode = '404'")
```

```
In [22]: consulta.take(10)
```

```
Out[22]: [Row(url='/pub/winvn/readme.txt', resultado=2004),
Row(url='/pub/winvn/release.txt', resultado=1732),
Row(url='/shuttle/missions/STS-69/mission-STS-69.html', resultado=682),
Row(url='/shuttle/missions/sts-68/ksc-upclose.gif', resultado=426),
Row(url='/history/apollo/a-001/a-001-patch-small.gif', resultado=384)]
```

```
In [23]: consulta.show()
```

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

0.1.7 Resposta:

```
/pub/winvn/readme.txt
/pub/winvn/release.txt
/shuttle/missions/STS-69/mission-STS-69.html
/shuttle/missions/sts-68/ksc-upclose.gif
/history/apollo/a-001/a-001-patch-small.gif
```

0.1.8 4) Quantidade de erros 404 por dia

In [24]: *# Executando SQL - Quantidade de erros 404 por dia*

```
sqlContext.sql("select Data, count(HttpStatusCode) as resultado from linhasTB where HttpStatusCode = 404")
```

```
+-----+-----+
|  Data|resultado|
+-----+-----+
|01/Aug|    242|
|01/Jul|    315|
|02/Jul|    291|
|03/Aug|    300|
|03/Jul|    473|
|04/Aug|    343|
|04/Jul|    355|
|05/Aug|    232|
|05/Jul|    492|
|06/Aug|    371|
|06/Jul|    633|
|07/Aug|    526|
|07/Jul|    568|
|08/Aug|    386|
|08/Jul|    302|
|09/Aug|    277|
|09/Jul|    342|
|10/Aug|    312|
|10/Jul|    392|
|11/Aug|    260|
+-----+-----+
```

only showing top 20 rows

0.1.9 Resposta:

```
+-----+-----+
|  Data|resultado|
+-----+-----+
|01/Aug|    242|
|01/Jul|    315|
|02/Jul|    291|
|03/Aug|    300|
|03/Jul|    473|
|04/Aug|    343|
|04/Jul|    355|
|05/Aug|    232|
|05/Jul|    492|
|06/Aug|    371|
```

```

|06/Jul| 633|
|07/Aug| 526|
|07/Jul| 568|
|08/Aug| 386|
|08/Jul| 302|
|09/Aug| 277|
|09/Jul| 342|
|10/Aug| 312|
|10/Jul| 392|
|11/Aug| 260|
+-----+
only showing top 20 rows

```

0.1.10 5) O total de bytes retornados

```
In [25]: data_df.groupby().sum().collect()[0][0]
```

```
Out[25]: 65401233313
```

0.1.11 Resposta: 65401233313 bytes

```
In [26]: # Executando SQL com a Resposta da Soma Total de Bytes
sqlContext.sql("select sum(Bytes) as resultado from linhasTB").show()
```

```

+-----+
| resultado|
+-----+
|65401233313|
+-----+

```

```
In [27]: # Limpando a memória
del joinedLinhasL, data_df
```

0.1.12 PROBLEMAS NAS CONVERSÕES DE LINHAS

Necessário análise do arquivo para verificar os motivos e as inconsistências. Foi decidido por mim, nesse desafio ignorar essas linhas e processar as demais.

```
In [28]: sqlContext.sql("select count(*) as erros from linhasTB WHERE ErroConversaoBytes > 0 or
```

```

+-----+
|erros|
+-----+
| 6539|
+-----+

```

0.1.13 Resposta: 6539 erros na conversão

0.1.14 Percentual de erros na conversao => (6539 / 3461613) * 100 => 0.18%

```
In [29]: sqlContext.sql("select * from linhasTB WHERE ErroConversaoBytes > 0 or ErroConversaoData > 0 or ErroConversaoHttp > 0 or ErroConversaoUrl > 0")
```

Bytes	Data	ErroConversaoBytes	ErroConversaoData	ErroConversaoHttp	ErroConversaoUrl
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0
0	01/Jul	1	0	1	0

only showing top 20 rows

```
In [30]: #Fechando o contexto
         sc.stop()
```

0.2 FIM

0.3 OBRIGADO

```
In [ ]:
```