HTTP requests to the NASA Kennedy Space Center WWW server

Code

Análise exploratória da base de dados que possui requisições HTTP para o servidor NASA Kennedy Space Center na Flórida, no período de Julho de 1995 a Agosto de 1995. Os arquivos originais podem ser obtidos em http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html (http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html). A exploração será feita utilizando o Apache Spark e a linguagem R com a biblioteca “Sparklyr”

A base original é disponibilizada em dois arquivos chamados “access_log_Jul95” e “access_log_Aug95”. Será criado um novo arquivo chamado “nasa_logs.tsv” a partir da concatenação dos dois arquivos originais. Essa junção é feita executando o seguinte comando no shell do Linux:

```
$cat access_log_Jul95 access_log_Aug95 > nasa_logs.tsv
```

A base de dados contém as seguintes informações:

- **Host** fazendo a requisição;
- **Timestamp** da requisição no formato “DIA/MÊS/ANO:HH:MM:SS TIMEZONE”;
- **Requisição**, entre aspas;
- **Código de retorno HTTP**
- **Total de bytes retornados**

Os dados originais apresentam o seguinte formato:

```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
```

Para facilitar a manipulação dos dados o arquivo será transformado para o formato tsv, ou seja, os valores serão separados por TAB (\t). Para fazer essas tranformações, são executados os seguintes comandos no shell do Linux:

```
# Coloca \t entre o valor do host e da timestamp
$sed -Ei 's/\s-\s-\s\[/\t/' nasa_logs.tsv

# Coloca \t entre o valor da timestamp e da requisição
sed -Ei 's/\] \"/\t\"/' nasa_logs.tsv

# Coloca \t entre o valor da requisição e do código de retorno
sed -Ei 's/\" ([0-9]{3})/\"\t\1/' nasa_logs.tsv

# Coloca \t entre o valor do código de retorno e os bytes retornados
sed -Ei 's/([0-9]{3}) /\1\t/' nasa_logs.tsv
```

Assim, o arquivo pode ser carregado para o Spark utlizando a biblioteca Sparklyr

Hide

```
# Carregamento das bibliotecas
library(sparklyr)
library(dplyr)
# Criação do contexto Spark
sc <- spark_connect(master = "local", spark_home = "/home/filipe/spark-2.3.1-bin-hadoop2.7/")
```

```
Re-using existing Spark connection to local
```

Hide

```
# Carregamento da base
logs_spk = spark_read_csv(sc, "nasa_logs", "/home/filipe/Downloads/nasa_logs.tsv",
                          header = FALSE, columns = c("host", "timestamp", "request",
                                                      "reply_code", "total_bytes"),
                          delimiter = "\t")
# Alguns ajustes no dataset
logs_spk = logs_spk %>% mutate(total_bytes = regexp_replace(total_bytes, "-", "0")) # troca traços no total de by
tes pelo número 0
logs_spk = logs_spk %>% mutate(total_bytes = as.integer(total_bytes)) # Converte o total de bytes para inteiro
logs_spk = logs_spk %>% mutate(timestamp = regexp_replace(timestamp, ":.*", "")) # Retira as informações de hora
 da timestamp
# Preview do dataset
head(logs_spk)
```

host <chr>	**timestamp** <chr>	**request** <chr>
199.72.81.55	01/Jul/1995	GET /history/apollo/ HTTP/1.0
unicomp6.unicomp.net	01/Jul/1995	GET /shuttle/countdown/ HTTP/1.0
199.120.110.21	01/Jul/1995	GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0
burger.letters.com	01/Jul/1995	GET /shuttle/countdown/liftoff.html HTTP/1.0
199.120.110.21	01/Jul/1995	GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0
burger.letters.com	01/Jul/1995	GET /images/NASA-logosmall.gif HTTP/1.0

6 rows | 1-3 of 5 columns

Número de hosts únicos

O número de hosts únicos podem ser obtidos da seguinte forma

Hide

```
unique_host = logs_spk %>% group_by(host) %>% summarise(count = n()) %>%  arrange(desc(count))
count(unique_host)
```

n
<dbl>
137979

1 row

Há um total de 137979 hosts diferentes. Podemos ver também quais os hosts com mais acessos

Hide

```
head(unique_host)
```

host	count
<chr>	<dbl>
piweba3y.prodigy.com	21988
piweba4y.prodigy.com	16437
piweba1y.prodigy.com	12825
edams.ksc.nasa.gov	11964
163.206.89.4	9697
news.ti.com	8161

6 rows

Total de erros 404

O total de erros 404 pode ser obtido da seguinte forma

Hide

```
unique_reply = logs_spk %>% group_by(reply_code) %>% summarise(count = n()) %>%  arrange(desc(count))
head(unique_reply)
```

reply_code <int>	count <dbl>
200	3100522
304	266773
302	73070
404	20901
403	225
500	65

6 rows

Podemos ver que 20901 requisições tiveram erro 404. O código de retorno mais frequente foi 200, ocorrendo mais de 3,1 milhões de vezes.

Os 5 URLs que mais causaram erro 404

Essa informação é obtida através do seguinte comando:

Hide

```
url_404 = logs_spk %>% filter(reply_code == "404") %>% group_by(host) %>% summarise(count = n()) %>%  arrange(des
c(count))
head(url_404)
```

host <chr>	count <dbl>
hoohoo.ncsa.uiuc.edu	251
piweba3y.prodigy.com	157
jbiagioni.npt.nuwc.navy.mil	132

host <chr>	count <dbl>
piweba1y.prodigy.com	114
www-d4.proxy.aol.com	91
piweba4y.prodigy.com	86

6 rows

Quantidade de erros 404 por dia

De maneira análoga, podemos obter a quantidade de erros 404 por dia

Hide

```
days_404 = logs_spk %>% filter(reply_code == "404") %>% group_by(timestamp) %>% summarise(count = n()) %>%  arran
ge(desc(count))
head(days_404)
```

timestamp <chr>	count <dbl>
06/Jul/1995	640
19/Jul/1995	639
30/Aug/1995	571
07/Jul/1995	570
07/Aug/1995	537
13/Jul/1995	532

6 rows

O dia com mais erros 404 foi dia 6 de julho com 640 ocorrências. O dia 19 de julho vem logo em seguida, com 639. No mês de Agosto, o dia com mais ocorrência de 404 foi dia 30 com 571.

Total de bytes retornados

Hide

```
bytes = logs_spk %>% select(total_bytes) %>% summarise(total = sum(total_bytes))
bytes
```

total
<dbl>
65524314915

1 row

Foram retornados 65524314915 bytes, que equivale a 65,5GB