

## Tutorial 02 – Exibição de Grandes Volumes de Dados

### Cenário

**Seu Gerente:** é indiferente, você produziu o que você sempre produziu - um relatório sobre dados estruturados, mas você realmente não provou nenhum valor adicional

**Você:** concordo e apenas voltarei a fazer o que sempre foi feito e esquece daquele aumento ... ou você tem um ás na manga ...

### Correlacionando dados estruturados com dados não estruturados

Como você é uma pessoa de dados bastante inteligente, você percebe que outra questão comercial interessante seria: **os produtos mais vistos também são os mais vendidos?** (ou para outros cenários, os mais procurados, os mais falados sobre ...). Uma vez que a *Hadoop* pode armazenar dados não estruturados e semiestruturados juntamente com dados estruturados sem remodelar um banco de dados inteiro, você também pode ingerir, armazenar e processar **eventos de log da web**. Vamos descobrir o que os visitantes do site realmente viram.

Para isso, você precisa dos dados do clique da web. A maneira mais comum de obter estes dados é com o *clickstream* da web é usando o [Apache Flume](#). O *Flume* é um *framework* escalável que obtêm em tempo real dados que permitem rotear, filtrar, agregar e fazer "mini-operações" nos dados para a plataforma de processamento escalável.

Mais adiante neste tutorial, você poderá explorar um exemplo de configuração do *Flume*, para usar para obter e transformar em tempo real os nossos dados de fluxo de dados da amostra. No entanto, por causa do tempo do tutorial, nesta etapa, não teremos a paciência de aguardar três dias para obter os dados. Em vez disso, foi preparado um conjunto de dados da web do clickstream (com o objetivo de avançar rapidamente três dias) para que você possa fazer *upload* em massa no *HDFS* diretamente.

### Dados de Upload em Massa

Por conveniência, carregamos uma amostra (cerca de 180K linhas) de um mês de dados de log de acesso em **/opt/examples/log\_files/access.log.2**.

Vamos mover esses dados do sistema de arquivos local para o HDFS.

Volte para o seu terminal e execute os seguintes comandos do Nó do Gerenciador.

```
sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse/original_access_logs
sudo -u hdfs hadoop fs -copyFromLocal /opt/examples/log_files/access.log.2
/user/hive/warehouse/original_access_logs
```

O comando de cópia pode levar um bom tempo para ser concluído.

Você deve ver um resultado semelhante ao seguinte:



```
cloudera@quickstart:~$
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/original_access_logs
Found 1 items
-rw-r--r-- 1 hdfs supergroup 39593868 2017-11-22 08:51 /user/hive/warehouse/original_access_logs/access.log.2
[cloudera@quickstart ~]$
```

Agora, você pode construir uma tabela na **Hive** e consultar os dados via **Impala** e **Hue**. Você irá construir esta **tabela em 2 etapas**. Em primeiro lugar, você aproveitará o **SerDe** (**Serizadores** / **Deserializadores**) flexíveis da **Hive** para analisar os logs em campos individuais usando uma **expressão regular**. Em segundo lugar, você transferirá os dados dessa tabela intermediária para um que não requer qualquer **SerDe** especial. Uma vez que os dados estão nesta tabela, você pode consultá-lo muito mais rápido e de forma mais interativa usando o **Impala**.

Usaremos o aplicativo **Hive Query Editor** em **Hue** para executar as seguintes consultas:

```
CREATE EXTERNAL TABLE intermediate_access_logs (  
  ip STRING,  
  date STRING,  
  method STRING,  
  url STRING,  
  http_version STRING,  
  code1 STRING,  
  code2 STRING,  
  dash STRING,  
  user_agent STRING)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'  
WITH SERDEPROPERTIES (  
  'input.regex' = '([^ ]*) - - \\[([^\]]*)\\] \"([^\"]*) ([^\"]*) ([^\"]*)\" (\\d*)  
(\\d*) \"([^\"]*)\" \"([^\"]*)\"',  
  'output.format.string' = \"%1$$$$ %2$$$$ %3$$$$ %4$$$$ %5$$$$ %6$$$$ %7$$$$  
%8$$$$ %9$$$$\")  
LOCATION '/user/hive/warehouse/original_access_logs';  
  
CREATE EXTERNAL TABLE tokenized_access_logs (  
  ip STRING,  
  date STRING,  
  method STRING,  
  url STRING,  
  http_version STRING,  
  code1 STRING,  
  code2 STRING,  
  dash STRING,  
  user_agent STRING)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION '/user/hive/warehouse/tokenized_access_logs';  
  
ADD JAR /usr/lib/hive/lib/hive-contrib.jar;  
  
INSERT OVERWRITE TABLE tokenized_access_logs SELECT * FROM  
intermediate_access_logs;
```

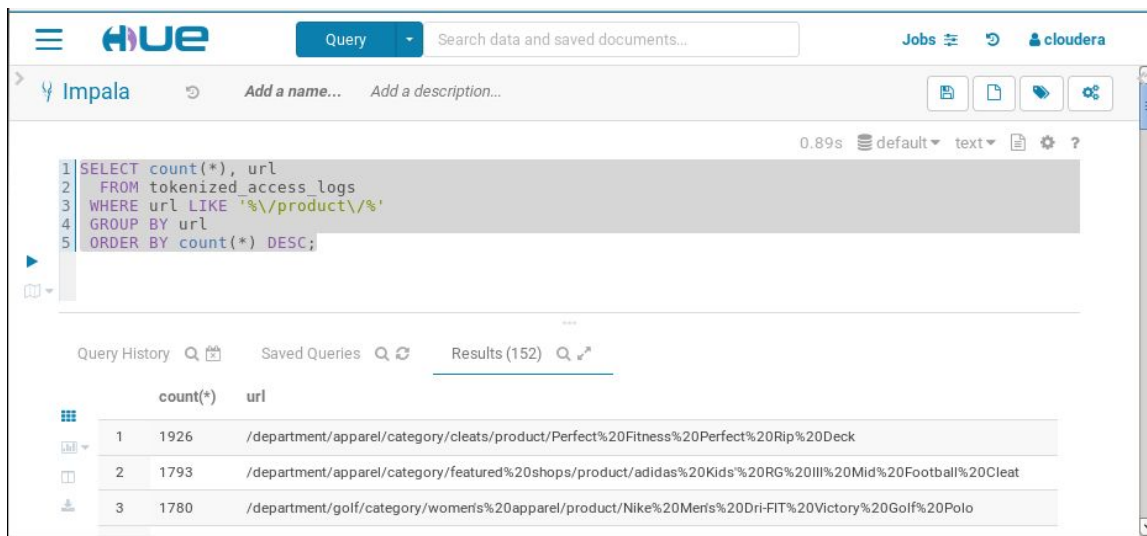
A consulta final levará um minuto para ser executada. Está usando um trabalho **MapReduce**, assim como a nossa importação do **Sqoop** para transferir os dados de uma tabela para a outra em paralelo. Novamente, precisamos dizer ao **Impala** que algumas tabelas foram criadas através de uma ferramenta diferente. Volte para o aplicativo **Impala Query Editor** e digite o seguinte comando:

```
invalidate metadata;
```

Agora, se você digitar "show tables;", poderá consultar ou atualizar a lista de tabelas na coluna da esquerda. Poderá ver as duas novas tabelas externas no banco de dados padrão. Cole a seguinte consulta no *Query Editor*:

```
SELECT count(*), url
FROM tokenized_access_logs
WHERE url LIKE '%\product\/%'
GROUP BY url
ORDER BY count(*) DESC;
```

Você deve ver um resultado semelhante ao seguinte:



The screenshot shows the Hue Query Editor interface. At the top, there's a search bar and a 'Query' dropdown. Below that, the 'Impala' tab is active. The SQL query is entered in the editor:

```
1 SELECT count(*), url
2 FROM tokenized_access_logs
3 WHERE url LIKE '%\product\/%'
4 GROUP BY url
5 ORDER BY count(*) DESC;
```

Below the query editor, the 'Results (152)' tab is selected, showing a table with two columns: 'count(\*)' and 'url'. The first three rows are visible:

	count(*)	url
1	1926	/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck
2	1793	/department/apparel/category/featured%20shops/product/adidas%20Kids%20RG%20III%20Mid%20Football%20Clea
3	1780	/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo

Ao analisar os resultados, irás perceber rapidamente que esta lista contém muitos dos produtos da lista mais vendida das etapas anteriores do tutorial 1, mas há um produto que não apareceu no resultado anterior. **Existe um produto (Deck) que foi visto muito, mas nunca foi comprado. Por quê?**

Bem, no nosso exemplo com *GoData*, uma vez que essas descobertas estranhas são apresentadas ao seu gerente, ele é imediatamente escalado. Eventualmente, alguém descobre que naquela página de visualização, onde a maioria dos visitantes parou, o caminho de vendas do produto teve um erro de digitação no preço do item. Uma vez que o erro de digitação foi corrigido e um preço correto foi exibido, as vendas desse item começaram a aumentar rapidamente.

## CONCLUSÃO

Se você não tivesse tido uma ferramenta eficiente e interativa que permita a análise de dados semi-estruturados de alto volume, essa perda de receita teria sido perdida por um longo período de tempo. Existe risco de perda se uma organização buscar respostas dentro de dados parciais. A correlação de dois conjuntos de dados para a mesma questão comercial mostrou valor, e ser capaz de fazê-lo dentro da mesma plataforma facilitou a vida para você e para a organização.