

[< Showing Big Data Value](#)[Advanced Analytics in the
Same Platform >](#)

Tutorial Exercise 3

Correlate Structured Data with Unstructured Data

Since you are a pretty smart data person, you realize another interesting business question would be: *are the most viewed products also the most sold?* (or for other scenarios, the most searched for, the most chatted about...). Since Hadoop can store unstructured and semi-structured data alongside structured data without remodelling an entire database, you can just as well ingest, store and process web log events. Let's find out what site visitors have actually viewed the most.

For this, you need the web clickstream data. The most common way to ingest web clickstream is to use Flume. Flume is a scalable real-time ingest framework that allows you to route, filter, aggregate, and do "mini-operations" on data on its way in to the scalable processing platform.

In Exercise 4, later in this tutorial, you can explore a Flume configuration example, to use for real-time ingest and transformation of our sample web clickstream data. However, for the sake of tutorial-time, in this step, we will not have the patience to wait for three days of data to be ingested. Instead, we prepared a web clickstream data set (just pretend you fast forwarded three days) that you can bulk upload into HDFS directly.

Bulk Upload Data

For convenience, we have loaded a sample (about 20MM lines) of one month's worth of access log data into `/opt/examples/log_data/access.log.2`.

Let's move this data from the local filesystem, into HDFS.

Go back to your terminal and execute the following commands from your **Master Node**.

[Copy](#)

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse/original_access_logs
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -copyFromLocal /opt/examples/log_files/access.log.2 /user/hive/warehouse/original_access_logs
```

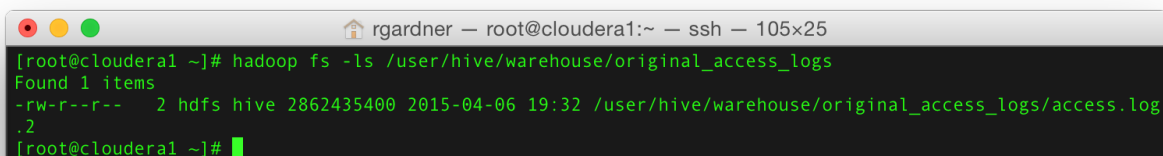
The copy command may take several minutes to complete.

Verify that your data is in HDFS by executing the following command:

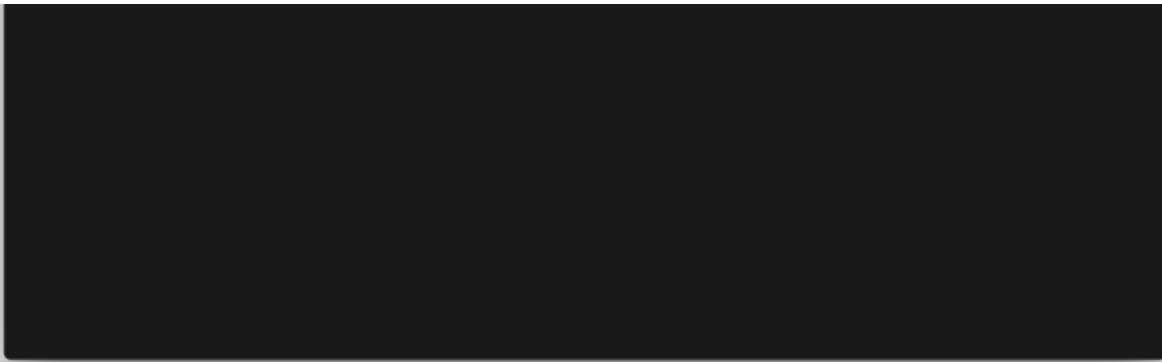
[Copy](#)

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/original_access_logs
```

You should see a result similar to the following:



```
rgardner — root@cloudera1:~ — ssh — 105x25
[root@cloudera1 ~]# hadoop fs -ls /user/hive/warehouse/original_access_logs
Found 1 items
-rw-r--r--  2 hdfs hive 2862435400 2015-04-06 19:32 /user/hive/warehouse/original_access_logs/access.log.2
[root@cloudera1 ~]#
```



Now you can build a table in Hive and query the data via Impala and Hue. You'll build this table in 2 steps. First, you'll take advantage of Hive's flexible SerDes (serializers / deserializers) to parse the logs into individual fields using a regular expression. Second, you'll transfer the data from this intermediate table to one that does not require any special SerDe. Once the data is in this table, you can query it much faster and more interactively using Impala.

We'll query Hive using a command-line JDBC client for Hive called Beeline. You can invoke it from the terminal with the following:

```
[cloudera@quickstart ~]$ beeline -u jdbc:hive2://quickstart:10000/default -n admin -d org.apache
```

Copy

Once the Beeline shell is connected, run the following queries:

[Copy](#)

```

0: jdbc:hive2://quickstart:10000/default> 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$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s\"
)
LOCATION '/user/hive/warehouse/original_access_logs';

0: jdbc:hive2://quickstart:10000/default> 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';

0: jdbc:hive2://quickstart:10000/default> ADD JAR /usr/lib/hive/lib/hive-contrib.jar;

0: jdbc:hive2://quickstart:10000/default> INSERT OVERWRITE TABLE tokenized_access_logs SELECT *

0: jdbc:hive2://quickstart:10000/default> !quit

```

To save time during queries, Impala does not poll constantly for metadata changes. So when you create new tables while Impala is running you must tell it to refresh the metadata. Go to Hue and open the Impala Query Editor app, and enter the following command:

[Copy](#)

```
invalidate metadata;
```

Now, if you run 'show tables' or refresh the table list in the left-hand column, you should see the two new external tables in the default database. Paste the following query into the Query Editor

[Copy](#)

```
select count(*),url from tokenized_access_logs
where url like '%\product\%'
group by url order by count(*) desc;
```

You should see a result similar to the following:

The screenshot shows the Hue Query Editor interface. On the left, there's a sidebar with 'Assist' and 'Settings' tabs. Under 'Assist', there's a 'DATABASE' section with a dropdown set to 'default' and a 'Table name...' input field. Below that, a list of database tables is shown: categories, customers, departments, order_items, orders, products, and tokenized_a... (truncated). The main area is titled 'Query Editor' and contains a SQL query:

```
1 select count(*),url from tokenized_access_logs
2 where url like '%\product\%'
3 group by url order by count(*) desc;
4
```

Below the query, there are buttons: 'Execute', 'Save as...', 'Explain', and 'or create a New query'. The 'Execute' button is highlighted. Below the buttons, there's a tabbed interface with 'Recent queries', 'Query', 'Log', 'Columns', 'Results', and 'Chart'. The 'Results' tab is selected, showing a table with 12 rows and 2 columns: 'count(*)' and 'url'. The first row has a count of 129996 and a URL starting with '/department/apparel/category/featured%20shops/product/adidas%20Kids%20RG%20II%20Mid%20Football%20C...'. The last row has a count of 64716 and a URL starting with '/department/fitness/category/lacrosse/product/Under%20Armour%20Men%20Tech%20II%20T-Shirt'.

If one of these steps fail, please reach out to our Cloudera Live Forum (<http://community.cloudera.com/t5/Cloudera-Live/bd-p/ClouderaLive>) and get help.

By introspecting the results you quickly realize that this list contains many of the products on the most sold list from previous tutorial steps, but there is one product that did not show up in the previous result. There is one product that seems to be viewed a lot, but never purchased. Why?

	product_id	product_name	revenue
0	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.282318115
1	365	Perfect Fitness Perfect Rip Deck	4233794.3682899475
2	957	Diamondback Women's Serene Classic Comfort BI	3946837.004547119
3	191	Nike Men's Free 5.0+ Running Shoe	3507549.2067337036

4	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
5	1073	Pelican Sunstream 100 Kayak	2967851.6815185547
6	1014	O'Brien Men's Neoprene Life Vest	2765543.314743042
7	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.4868011475
8	627	Under Armour Girls' Toddler Spine Surge Runni	1214896.220287323
9	565	adidas Youth Germany Black/Red Away Match Soc	63490

Recent queriesQueryLogColumnsResultsChart

count(*)url

0248650/department/apparel/category/featured%20shops/product/adidas%20Kids%20RG%20III%20Mid%20Football%20CleatMissing???

1248128/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck2nd

2247914/department/apparel/category/men's%20footwear/product/Nike%20Men's%20CJ%20Elite%202%20TD%20Football%20Cleat8th

3247456/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo5th

4149182/department/fan%20shop/category/indoor/outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest7th

5148995/department/fan%20shop/category/fishing/product/Field%20&%20Stream%20Sportsman%2016%20Gun%20Fire%20Safe1st!

6148495/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak6th

7147921/department/fan%20shop/category/camping%20&%20hiking/product/Diamondback%20Women's%20Serene%20Classic%20Comfort%20Bi3rd

8124969/department/footwear/category/cardio%20equipment/product/Nike%20Men's%20Free%205.0+%20Running%20Shoe4th

9124555/department/golf/category/shop%20by%20sport/product/Under%20Armour%20Girls%20Toddler%20Spine%20Surge%20Runni9th

Well, in our example with DataCo, once these odd findings are presented to your manager, it is immediately escalated. Eventually, someone figures out that on that view page, where most visitors stopped, the sales path of the product had a typo in the price for the item. Once the typo was fixed, and a correct price was displayed, the sales for that SKU started to rapidly increase.

CONCLUSION

If you hadn't had an efficient and interactive tool enabling analytics on high-volume semi-structured data, this loss of revenue would have been missed for a long time. There is risk of loss if an organization looks for answers within partial data. Correlating two data sets for the same business question showed value, and being able to do so within the same platform made life easier for you and for the organization.

< Showing Big Data Value

Advanced Analytics in the
Same Platform >

© 2015 Cloudera (<http://www.cloudera.com>), Inc. All rights reserved | Terms & Conditions (<http://www.cloudera.com/content/cloudera/en/terms-of-service.html>) | Privacy Policy (<http://www.cloudera.com/content/cloudera/en/privacy-policy.html>)

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation.