CASE STUDY: ANALYZING NASA WEBSERVER LOGS WITH APACHE HIVE

Server logs contain lots of information from web servers, application logs, user-generated. This case study will show you how to derive insights from the web server logs. The insights can be used for monitoring servers, user behavior, fraud detection, improving business intelligence etc.

Download dataset

http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

File to be downloaded

Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed, 167.8 MB uncompressed.

Before moving to the activity, please go through the HTTP response codes at *https://en.wikipedia.org/wiki/List_of_HTTP_status_codes*

Login to *cloudxlab* and go to *Web console*.

After logging in, type '*Hive*' to start hive in web console.

Understanding Data

The dataset contains two month's worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The log files are stored in Apache Common Log Format (CLF).

host, identity, useridentity, time, request, status, size

in24.inetnebr.com - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0" 200 1839

Apache Common Log Format is made up of the following

1) host making the request. A hostname when possible, otherwise the Internet address if the name could not be looked up.

2) '-' user identity from remote machine, currently unavailable

3) '-' user identity from local machine, currently unavailable

4) timestamp in the format "DAY MON DD HH:MM:SS YYYY", where DAY is the day of the week, MON is the name of the month, DD is the day of the month, HH:MM:SS is the time of

day using a 24-hour clock, and YYYY is the year. The timezone is -0400

5) Request sent to the server

6) HTTP reply code

7) Size (bytes) in the of the file received

Activity

1) Create a table in database and load the data

DROP TABLE IF EXISTS nasa_log;

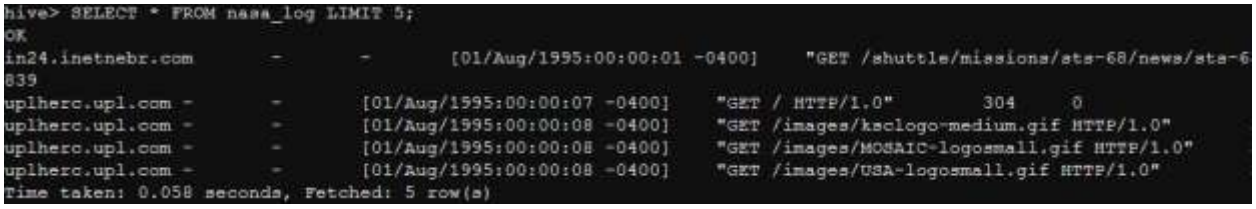
CREATE TABLE IF NOT EXISTS nasa_log (host String, identity String, userIdentity String, time String, request String, status String, size String) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe' WITH SERDEPROPERTIES ("input.regex" = "([^]) ([^]*) ([^]*) (-|\\[[^\\]]*\\]) ([^ \"]*|\"[^\"]*\") (-|[0-9]*) (-|[0-9]*)" , "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s") STORED AS TEXTFILE;*

2) Load data from storage

LOAD DATA INPATH '/user/raghavendrapruthvin9629/NASA/nasa_logs' OVERWRITE INTO TABLE nasa_log;

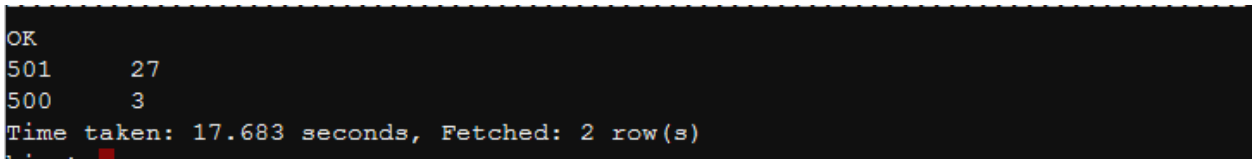
describe nasa_log;

*SELECT * FROM nasa_log LIMIT 5;*



3) Find the top endpoints that received server side error

SELECT status, count(request) FROM nasa_log GROUP BY status HAVING status == regexp_extract(status, '^50.', 0) ORDER BY status DESC LIMIT 5;



Refrences on *regexp_extract* here - *https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF*

4) Which resource is requested most frequently by the hosts

SELECT request,count(*) AS requestCount FROM nasa_log GROUP BY request ORDER BY requestCount DESC LIMIT 30;

```
OK
"GET /images/NASA-logosmall.gif HTTP/1.0"       96841
"GET /images/KSC-logosmall.gif HTTP/1.0"        75132
"GET /images/MOSAIC-logosmall.gif HTTP/1.0"     66966
"GET /images/USA-logosmall.gif HTTP/1.0"        66594
"GET /images/WORLD-logosmall.gif HTTP/1.0"      65975
"GET /images/ksclogo-medium.gif HTTP/1.0"       62293
"GET /ksc.html HTTP/1.0"        43379
"GET /history/apollo/images/apollo-logo1.gif HTTP/1.0"  37748
"GET /images/launch-logo.gif HTTP/1.0"  35059
"GET / HTTP/1.0"        29850
"GET /images/ksclogosmall.gif HTTP/1.0" 27758
"GET /shuttle/missions/sts-69/mission-sts-69.html HTTP/1.0"     24544
"GET /shuttle/countdown/ HTTP/1.0"      24335
"GET /shuttle/missions/sts-69/count69.gif HTTP/1.0"     24315
"GET /shuttle/missions/sts-69/sts-69-patch-small.gif HTTP/1.0"  23362
"GET /shuttle/missions/missions.html HTTP/1.0"  22339
"GET /images/launchmedium.gif HTTP/1.0" 19841
"GET /htbin/cdt_main.pl HTTP/1.0"       17179
"GET /shuttle/countdown/images/countclock.gif HTTP/1.0" 12128
"GET /icons/menu.xbm HTTP/1.0"  12125
"GET /icons/blank.xbm HTTP/1.0" 12043
"GET /icons/image.xbm HTTP/1.0" 10298
"GET /history/apollo/images/footprint-logo.gif HTTP/1.0"        10115
"GET /software/winvn/winvn.html HTTP/1.0"       10059
"GET /history/history.html HTTP/1.0"    10039
"GET /history/apollo/images/apollo-small.gif HTTP/1.0"  9391
"GET /history/apollo/images/footprint-small.gif HTTP/1.0"       9221
"GET /software/winvn/winvn.gif HTTP/1.0"        9023
"GET /history/apollo/apollo.html HTTP/1.0"      8957
```

5) Display the top 10 host who made maximum requests to the server

SELECT host, count(host) as topTenHost from nasa_log GROUP BY host ORDER BY topTenHost DESC LIMIT 10;

```
OK
edams.ksc.nasa.gov      6530
piweba4y.prodigy.com    4844
163.206.89.4    4791
piweba5y.prodigy.com    4607
piweba3y.prodigy.com    4416
www-d1.proxy.aol.com    3889
www-b2.proxy.aol.com    3534
www-b3.proxy.aol.com    3463
www-c5.proxy.aol.com    3423
www-b5.proxy.aol.com    3411
```

6) Find the total count of different response codes returned by the server

SELECT status, count(host) FROM nasa_log GROUP BY status;

```
OK
NULL    0
200     1398987
302     26497
304     134146
400     10
403     171
404     10039
500     3
501     27
```

7) How many hosts have accessed the server more than 1500 times

SELECT host,count(*) AS hostCount FROM nasa_log GROUP BY host HAVING hostCount > 1500;

```
OK
163.206.89.4    4791
edams.ksc.nasa.gov      6530
intgate.raleigh.ibm.com 3123
mpngate1.ny.us.ibm.net  3011
news.ti.com     3298
piweba3y.prodigy.com    4416
piweba4y.prodigy.com    4844
piweba5y.prodigy.com    4607
www-a1.proxy.aol.com    3041
www-a2.proxy.aol.com    3337
www-b2.proxy.aol.com    3534
www-b3.proxy.aol.com    3463
www-b4.proxy.aol.com    3293
www-b5.proxy.aol.com    3411
www-c1.proxy.aol.com    3177
www-c2.proxy.aol.com    3407
www-c3.proxy.aol.com    3272
www-c4.proxy.aol.com    3134
www-c5.proxy.aol.com    3423
www-c6.proxy.aol.com    3088
www-d1.proxy.aol.com    3889
www-d2.proxy.aol.com    3404
www-d3.proxy.aol.com    3296
www-d4.proxy.aol.com    3234
```

8) Find the average, maximum and minimum size of resource returned by the server

SELECT avg(size) FROM nasa_log;

```
OK
17244.974439471396
```

SELECT max(size) FROM nasa_log;

```
OK
99981
```

SELECT min(size) FROM nasa_log;

```
OK
-
```

9) Find the total number of unique host sending request to server

SELECT count(DISTINCT host) FROM nasa_log;