International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 6, Issue 11
November 2017

Weblog and Social Networking Data Analysis Fostering To the Todays Business Using Hadoop Ecosystem

Mohammed Faisal, Sanjay Gupta
Department of Information Technology, Nizwa College of Technology

Introduction: *Big data has enhanced the business value, make a better advertisement, make better recommendation and get more user attraction, Big data has given a new definition to enhance business in the hi-tech era along with the lot of online data, we can have lot of offline data which can be used for analysis which could be predictive, prescription, diagnostic and descriptive. When the streaming data is becoming big, its sounds like more complicated, difficult to move, difficult to manage. But the technology simplified it that is nothing but Big Data which is really easy to handle, and from large volume of data we can do analysis by using the Hadoop ecosystem. In this paper will emphasize of predictive analysis using twitter data and weblog data using Hadoop, flume and hive. With the twitter we have discussed opinion mining, and with weblog data we have discussed how to collect the weblog data and how it is use full from business perspective. In this paper priority given to practically generated output along with justification and assumption, it's an effort to show the output practically in the relevant components of Hadoop Ecosystem.*

Keywords: Weblog, social networking data, twitter, twitter application, big data, Hadoop, Flume, and Hive.

1.1 Technological Background

Hadoop Ecosystem

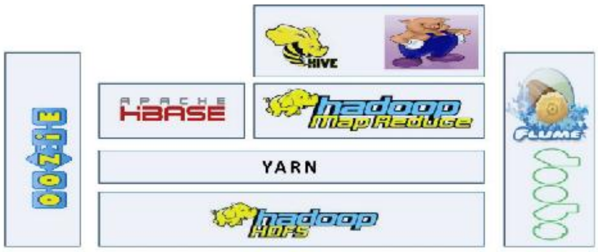


Fig 1 – Hadoop Ecosystem

Fig 1 is the illustration of Hadoop ecosystem, where HDFS is the bottom layer of ecosystem which managing the data in the Hadoop, just above of the HDFS we have YARN, which is providing the daemons, to do any kind of job we must start all daemons as a service, yarn has introduced in the Hadoop 2, and all other Hadoop tools name as HBASE and Hive is running over the yarn where MapReduce is the inbuilt component of Hadoop and HIVE is the top most components of ecosystem which having all the capabilities of all components which is placed below of hive.

In ecosystem, flume is most important component which is used to get the input file from the web, such as weblog files and Twitter data. Oozie is a workflow scheduler system to manage Apache Hadoop jobs.

In this paper we will through two examples which is really the hottest buzz in the industries from business point of view [1].

2.0 Opinion mining using Twitter application:

Opinion mining is the analysis of people's opinions, sentiments, evaluations, appraisals, attitudes and emotions in relation to entities like individuals, products, events, services, organizations and topics by classifying the expressions as negative / positive opinions [2].

With this example of opinion mining from business perspective, we can do analysis on any product to know the opinion of the users/people of the product. If opinion if positive we can increase the production, if the opinion is negative we can do the changes into the product or we can reduce/stop the production. In this way it will help the merchandisers which will definitely solid and strait factors to predict the demand of the product into the current market.

2.1 Scope of work:

In this example you will be trying to do opinion mining of the opinions on twitter using

Hadoop Ecosystem. The reason we chose twitter is because the social media is gaining popularity for the customer reviews and it is also creating a good business, customer relationship in the market. The reviews would certainly reflect the service of the company in the market.

2.2 Objectives:

The objective of the example is to find the sentiments of the users on Twitter. Companies can analyze the

Sentiments of the users regarding their products / services and use it for betterment of the same.

2.3 Plan of work:

i. **Collect Data**: The initial step is to collect all twitter tweets.
ii. **Preprocess Data:** We need to write MR Job to preprocess the data.
iii. **Classification of Data:** Write Hive UDF for the classification of the data into positive / negative Opinions.
iv. **Print Data:** Final stage would be to print the desired results with the number of good and bad Tweets collected.

2.3.1 Collect Data

Step 1: To collect the data from the twitter first we need to create an application from https://apps.twitter.com/ and we must generate the 2 types of API keys and 2 kind's access token as the following:

Consumer Key (API Key):
**********TD1sAyfQMMNIkh43Zrr

Consumer Secret (API Secret):
*************U5ycG5BAzlA1w4HjSZRGJaw7RuwMk5fcoW7AuEBLK

Access Token: ***********4082176-VUAHhOS0dGVlkUljIartLy1qlqqAg2n

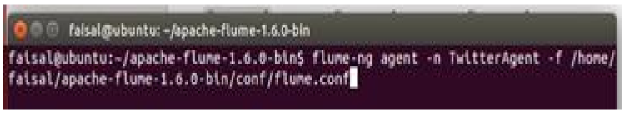
Access Token Secret:
*******8fGsJvvuJ0F47m4nmo2MlxCaPawg6dk3HjC

The above keys and access token should be updated into the flume configuration file.

Step 2: Create a Configuration file flume.conf file under **apache-flume-1.6.0-bin/conf** folder as following:

```
TwitterAgent.sources= Twitter
TwitterAgent.channels= MemChannel
TwitterAgent.sinks=HDFS
TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels=MemChannel
TwitterAgent.sources.Twitter.consumerKey= ***********TD1sAyfQMMNIkh43Zrr
TwitterAgent.sources.Twitter.consumerSecret=**********U5ycG5BAzlA1w4HjSZRGJaw7RuwMk5fcoW7AuEBLK
TwitterAgent.sources.Twitter.accessToken=***********4082176-VUAHhOS0dGVlkUljIartLy1qlqqAg2n
TwitterAgent.sources.Twitter.accessTokenSecret= *********8fGsJvvuJ0F47m4nmo2MlxCaPawg6dk3HjC
TwitterAgent.sources.Twitter.keywords= Toyota
TwitterAgent.sinks.HDFS.channel=MemChannel
TwitterAgent.sinks.HDFS.type=hdfs
TwitterAgent.sinks.HDFS.hdfs.path=hdfs://localhost:9000/Faisal/Toyota
TwitterAgent.sinks.HDFS.hdfs.fileType=DataStream
TwitterAgent.sinks.HDFS.hdfs.writeformat=Text
TwitterAgent.sinks.HDFS.hdfs.batchSize=1000
TwitterAgent.sinks.HDFS.hdfs.rollSize=0
TwitterAgent.sinks.HDFS.hdfs.rollCount=10000
TwitterAgent.sinks.HDFS.hdfs.rollInterval=600
TwitterAgent.channels.MemChannel.type=memory
TwitterAgent.channels.MemChannel.capacity=10000
TwitterAgent.channels.MemChannel.transactionCapacity=1000
```

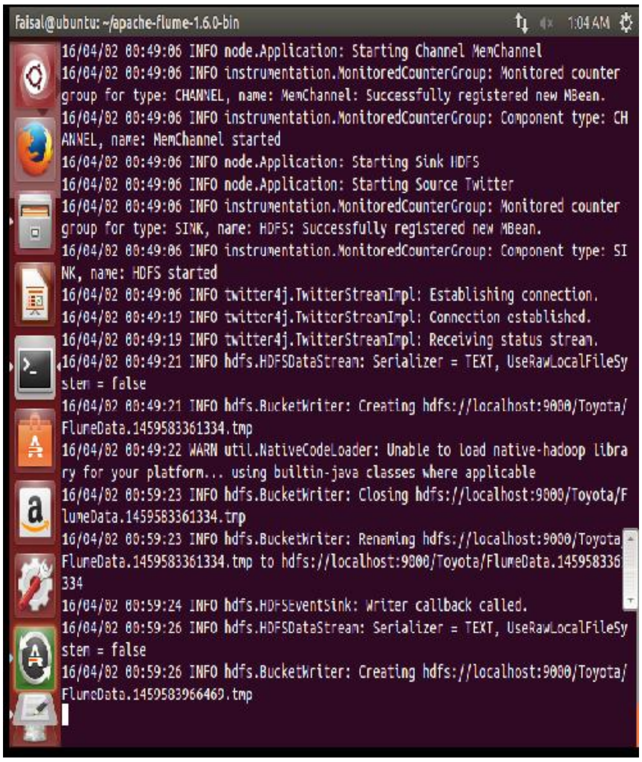
Step 3: Run below mentioned command to get the twitter data from local server as per given keywords, in this example we just given Toyota as keyword for the simplicity, keywords can be many as per the demand of analysis.



```
faisal@ubuntu: ~/apache-flume-1.6.0-bin
faisal@ubuntu:~/apache-flume-1.6.0-bin$ flume-ng agent -n TwitterAgent -f /home/
faisal/apache-flume-1.6.0-bin/conf/flume.conf
```

Fig 2: Snapshot to run the command to fetch the data from twitter

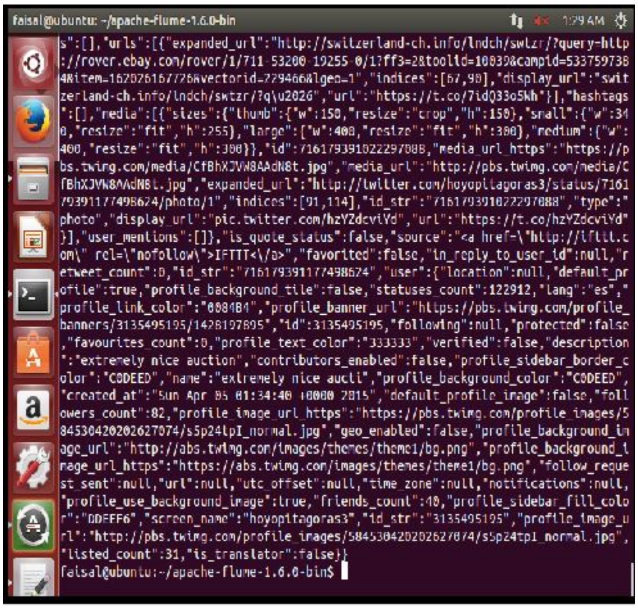
```
flume-ng agent -n TwitterAgent –f/home/faisal/apache-flume-1.6.0-bin/conf/flume.con
```



```
faisal@ubuntu: ~/apache-flume-1.6.0-bin                                1:04 AM
16/04/02 00:49:06 INFO node.Application: Starting Channel MemChannel
16/04/02 00:49:06 INFO instrumentation.MonitoredCounterGroup: Monitored counter
group for type: CHANNEL, name: MemChannel: Successfully registered new MBean.
16/04/02 00:49:06 INFO instrumentation.MonitoredCounterGroup: Component type: CH
ANNEL, name: MemChannel started
16/04/02 00:49:06 INFO node.Application: Starting Sink HDFS
16/04/02 00:49:06 INFO node.Application: Starting Source Twitter
16/04/02 00:49:06 INFO instrumentation.MonitoredCounterGroup: Monitored counter
group for type: SINK, name: HDFS: Successfully registered new MBean.
16/04/02 00:49:06 INFO instrumentation.MonitoredCounterGroup: Component type: SI
NK, name: HDFS started
16/04/02 00:49:06 INFO twitter4j.TwitterStreamImpl: Establishing connection.
16/04/02 00:49:19 INFO twitter4j.TwitterStreamImpl: Connection established.
16/04/02 00:49:19 INFO twitter4j.TwitterStreamImpl: Receiving status stream.
16/04/02 00:49:21 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSy
stem = false
16/04/02 00:49:21 INFO hdfs.BucketWriter: Creating hdfs://localhost:9000/Toyota/
FlumeData.1459583361334.tmp
16/04/02 00:49:22 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
16/04/02 00:59:23 INFO hdfs.BucketWriter: Closing hdfs://localhost:9000/Toyota/F
lumeData.1459583361334.tmp
16/04/02 00:59:23 INFO hdfs.BucketWriter: Renaming hdfs://localhost:9000/Toyota
FlumeData.1459583361334.tmp to hdfs://localhost:9000/Toyota/FlumeData.145958336
334
16/04/02 00:59:24 INFO hdfs.HDFSEventSink: Writer callback called.
16/04/02 00:59:26 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSy
stem = false
16/04/02 00:59:26 INFO hdfs.BucketWriter: Creating hdfs://localhost:9000/Toyota/
FlumeData.1459583966469.tmp
```

Fig 3: Snapshot to show the output of the snapshot Fig 2

We will get **FlumeData.1459583966469** data (Continuous similar files will get from nearest server) in the **Toyoto** directory in the **HDFS**. The contained of the files will be raw data which we need to parse by using Hive for further analysis purpose.



```
faisal@ubuntu: ~/apache-flume-1.6.0-bin                                1:29 AM
s":[],"urls":[{"expanded_url":"http://switzerland-ch.info/lndch/swlzr/?query-http
://rover.ebay.com/rover/1/711-53200-19255-0/1?ff3=2&toolid=10039&campid=53375973B
4&item=16202616772б&vectorid=229466&lgeo=1","indices":[67,90],"display_url":"swit
zerland-ch.info/lndch/swtzr/?q\u2026","url":"https://t.co/7idQ33o5Wh"}],"hashtags
":[],"media":[{"sizes":{"thumb":{"w":150,"resize":"crop","h":150},"small":{"w":34
0,"resize":"fit","h":255},"large":{"w":400,"resize":"fit","h":300},"medium":{"w":
400,"resize":"fit","h":300}},"id":716179391022297088,"media_url_https":"https://p
bs.twimg.com/media/CfBhXJVW8AAdN8t.jpg","media_url":"http://pbs.twimg.com/media/C
fBhXJVW8AAdN8t.jpg","expanded_url":"http://twitter.com/hoyopitagoras3/status/7161
79391177498624/photo/1","indices":[91,114],"id_str":"716179391022297088","type":"
photo","display_url":"pic.twitter.com/hzYZdcviYd","url":"https://t.co/hzYZdcviYd"
}],"user_mentions":[]},"is_quote_status":false,"source":"<a href=\"http://ifttt.c
om\" rel=\"nofollow\">IFTTT<\/a>","favorited":false,"in_reply_to_user_id":null,"r
etweet_count":0,"id_str":"716179391177498624","user":{"location":null,"default_pr
ofile":true,"profile_background_tile":false,"statuses_count":122912,"lang":"es","
profile_link_color":"0084B4","profile_banner_url":"https://pbs.twimg.com/profile_
banners/3135495195/1428197895","id":3135495195,"following":null,"protected":false
,"favourites_count":0,"profile_text_color":"333333","verified":false,"description
":"extremely nice auction","contributors_enabled":false,"profile_sidebar_border_c
olor":"C0DEED","name":"extremely nice aucti","profile_background_color":"C0DEED",
"created_at":"Sun Apr 05 01:34:40 +0000 2015","default_profile_image":false,"foll
owers_count":82,"profile_image_url_https":"https://pbs.twimg.com/profile_images/5
84530420202627074/s5p24tpI_normal.jpg","geo_enabled":false,"profile_background_im
age_url":"http://abs.twimg.com/images/themes/theme1/bg.png","profile_background_i
mage_url_https":"https://abs.twimg.com/images/themes/theme1/bg.png","follow_reque
st_sent":null,"url":null,"utc_offset":null,"time_zone":null,"notifications":null,
"profile_use_background_image":true,"friends_count":40,"profile_sidebar_fill_colo
r":"DDEEF6","screen_name":"hoyopitagoras3","id_str":"3135495195","profile_image_u
rl":"http://pbs.twimg.com/profile_images/584530420202627074/s5p24tpI_normal.jpg",
"listed_count":31,"is_translator":false}}
faisal@ubuntu:~/apache-flume-1.6.0-bin$
```

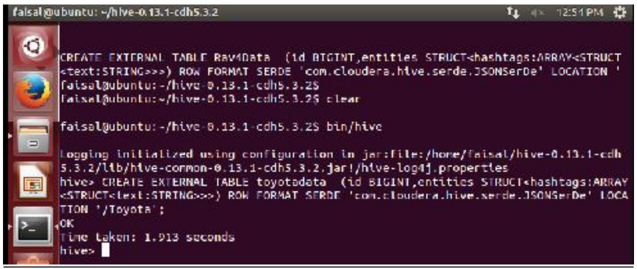
Fig4: Content of FlumeData.1459583966469

2.3.2 Preprocess Data

The above content is the raw data, it's in **JSON** format, and with this format we cannot analyze the data so we need to parse it with **HIVE.**

We need to create an external with the following command:

Step 1:CREATE EXTERNAL TABLE toyotadata (id BIGINT,entities STRUCT<hashtags:ARRAY<STRUCT<text:STRING>>>) ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe' LOCATION '/Toyota';

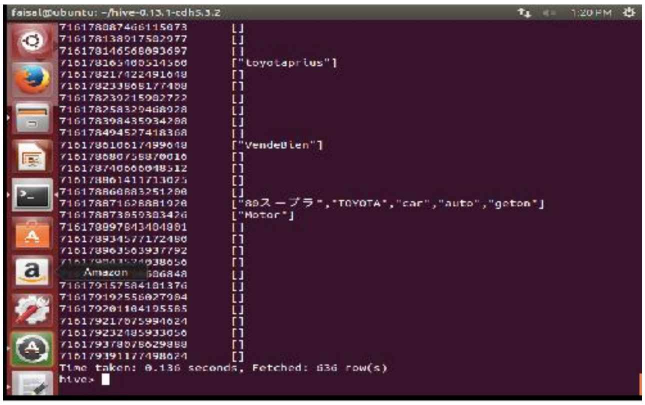


```
faisal@ubuntu: ~/hive-0.13.1-cdh5.3.2                                  12:51 PM

CREATE EXTERNAL TABLE Rav4Data  (id BIGINT,entities STRUCT<hashtags:ARRAY<STRUCT
<text:STRING>>>) ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe' LOCATION '
faisal@ubuntu:~/hive-0.13.1-cdh5.3.2$
faisal@ubuntu:~/hive-0.13.1-cdh5.3.2$ clear

faisal@ubuntu:~/hive-0.13.1-cdh5.3.2$ bin/hive

Logging initialized using configuration in jar:file:/home/faisal/hive-0.13.1-cdh
5.3.2/lib/hive-common-0.13.1-cdh5.3.2.jar!/hive-log4j.properties
hive> CREATE EXTERNAL TABLE toyotadata  (id BIGINT,entities STRUCT<hashtags:ARRAY
<STRUCT<text:STRING>>>) ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe' LOCA
TION '/Toyota';
OK
Time taken: 1.913 seconds
hive>
```

Fig5: Snapshot to create external table

to view the created table we need to run the following command:

Step 2: Select * from hashtoyota;



```
faisal@ubuntu: ~/hive-0.13.1-cdh5.3.2                                  1:20 PM
716178087466115073          []
716178138017582077          []
716178146568093697          []
716178163400514560          ["toyotaprius"]
716178217422491648          []
716178233868177408          []
716178239215982722          []
716178258329468928          []
716178398435034208          []
716178494527418368          []
716178610617499648          ["VendeBien"]
716178680758870016          []
716178740660048512          []
716178861411713025          []
716178860883251200          []
716178871628881920          ["86スープラ","TOYOTA","car","auto","geton"]
716178873850383426          ["Motor"]
716178897843484801          []
716178934577172480          []
716178963563937792          []
716179                 038656   []
          Amazon       586848   []
716179157584181376          []
716179192556027904          []
716179201184195585          []
716179217875994624          []
716179232485933056          []
716179378078629888          []
716179391177498624          []
Time taken: 0.136 seconds, Fetched: 636 row(s)
hive>
```

Fig5: Snapshot to view hashtoyota table

2.3.3 Classification of Data

Still we need to work out to break the data in the form of key, value where will break each key and its all correspondent values individually.

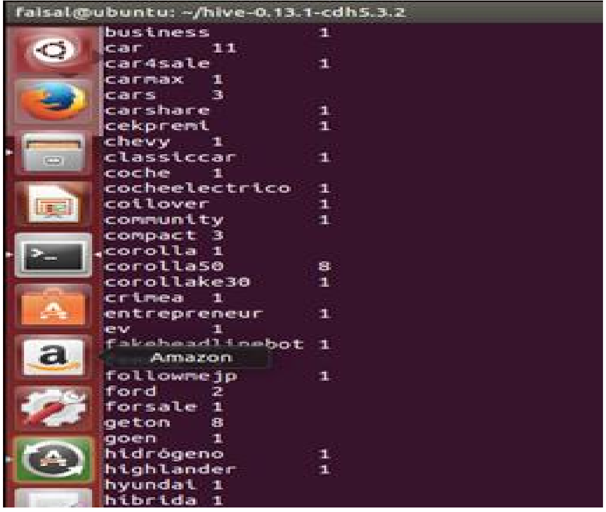
By running the following steps we can simplified the data which will be easy to analyze:

Step 1: create table hashtoyota_word as select id as id,hashtag from hashtoyota LATERAL VIEW explode(words) w as ht;

Step 2: select hashtag, count(hashtag) from hashtag_toyota group by hashtag;

2.3.4 Print Data

Finally we will get the data as follows:



```
faisal@ubuntu: ~/hive-0.13.1-cdh5.3.2
business        1
car     11
car4sale        1
carmax  1
cars    3
carshare        1
cekpremi        1
chevy   1
classiccar      1
coche   1
cocheelectrico  1
coilover        1
community       1
compact 3
corolla 1
corolla50       8
corollake30     1
crimea  1
entrepreneur    1
ev      1
fakeheadlinebot 1
followmejp      1
ford    2
forsale 1
geton   8
goen    1
hidrógeno       1
highlander      1
hyundai 1
hibrida 1
```

Fig 6: Snapshot to view hashtag_toyota counts of individual value of each key

With this format we can easily count the negative or positive verbs to get the conclusion, what is the opinion of people in that zone, because flume and twitter application will generate the data from the nearest server.

For the appropriate analysis as per the client requirements we can use various hive commands to customize our analysis.

3.0 Predictive analysis using weblog data of ecommerce sites:

After the emergence of ecommerce industry and rapid growth in this sector it is very important for the merchandiser to know every bit of information about the consumer like:

How many users hit every day to the websites?

From which regions more hits?

What type of product mostly they will look for?

User's feed back

And many more information.

Answers for the above questions can be answered if we able to analyses web log data efficiently.

To get complete information what is happening on the web site, the system should support gathering and processing information in **real-time**. Furthermore, as weblog data is collected and made available for analysis, it is also necessary to support **ad-hoc queries** via Apache Hive [3]

In Hadoop ecosystem there are many tools and technologies like Flume, Hive, Pig and Oozie which can be used for transporting and processing large amount of weblog data. In this section of the paper we will explore how we can transport weblogs data in HDFS using apache flume and then we will analyze it using apache Hive.

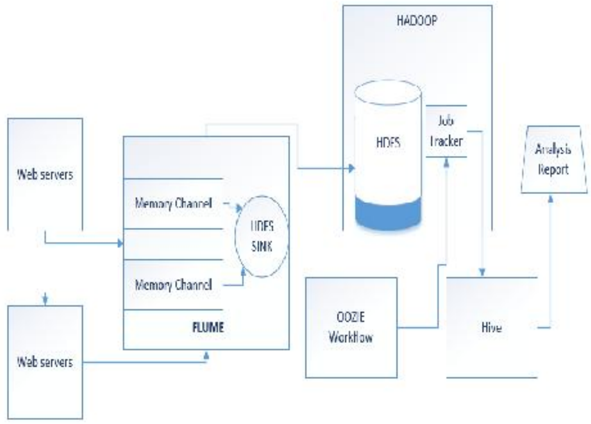
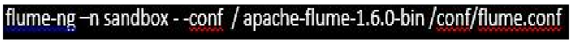


Fig 7: Illustration of conceptual view of weblog data analysis

Steps:

1. Transporting web logs data using Apache flume:

Apache flume allows users to stream data from multiple sources into Hadoop analysis, collect high volume weblogs in real time, guarantee data delivery and scale horizontally to handle additional data volume. Before configuring the flume we need to understand following components of flume:



```
flume-ng –n sandbox - -conf / apache-flume-1.6.0-bin /conf/flume.conf
```

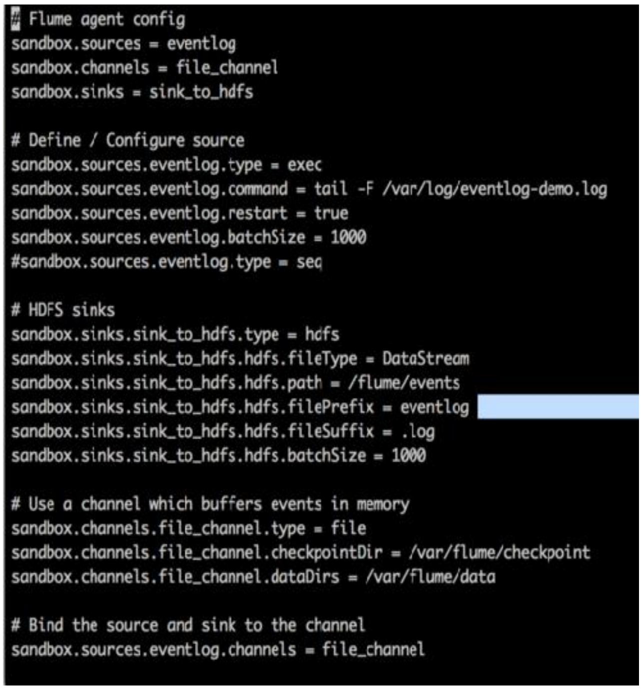
Source – the object through which data enters into Flume. Sources either actively poll for data or passively wait for data to be delivered to them. A variety of sources allow data to be collected, such as log4j logs.

Sink – the object that delivers the data to the destination. There many ways to design sink. One example is the HDFS sink that writes events to HDFS.

Channel – This act between the Source and the Sink. Sources ingest events into the channel and the sinks drain the channel.

Agent – any physical Java virtual machine running Flume. It is a collection of sources, sinks and channels. [4]

Now configure and start apache flume using following commands:



```
# Flume agent config
sandbox.sources = eventlog
sandbox.channels = file_channel
sandbox.sinks = sink_to_hdfs

# Define / Configure source
sandbox.sources.eventlog.type = exec
sandbox.sources.eventlog.command = tail -F /var/log/eventlog-demo.log
sandbox.sources.eventlog.restart = true
sandbox.sources.eventlog.batchSize = 1000
#sandbox.sources.eventlog.type = seq

# HDFS sinks
sandbox.sinks.sink_to_hdfs.type = hdfs
sandbox.sinks.sink_to_hdfs.hdfs.fileType = DataStream
sandbox.sinks.sink_to_hdfs.hdfs.path = /flume/events
sandbox.sinks.sink_to_hdfs.hdfs.filePrefix = eventlog
sandbox.sinks.sink_to_hdfs.hdfs.fileSuffix = .log
sandbox.sinks.sink_to_hdfs.hdfs.batchSize = 1000

# Use a channel which buffers events in memory
sandbox.channels.file_channel.type = file
sandbox.channels.file_channel.checkpointDir = /var/flume/checkpoint
sandbox.channels.file_channel.dataDirs = /var/flume/data

# Bind the source and sink to the channel
sandbox.sources.eventlog.channels = file_channel
```

Fig 8: Snapshot configuration file flume.conf file under apache-flume-1.6.0-bin/conf

Next we will configure **log4j.properties** and append the following in it.

```
flume, root, logger=INFO, LOGFILE
flume.log.dir=/var/log/flume
flume.log.file=flume.log
```

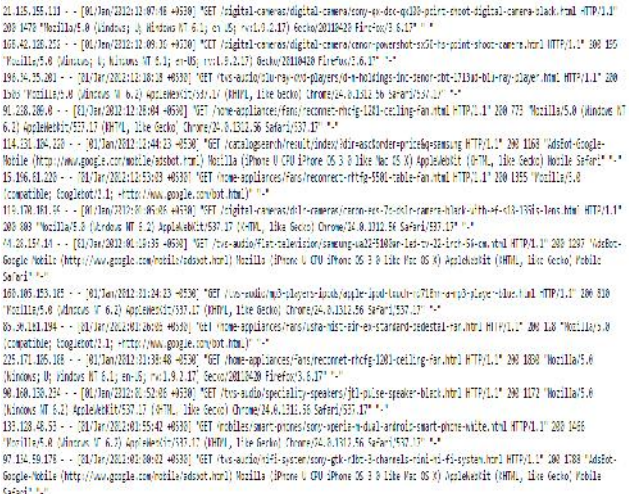
Now start flume using following command:

Now generate the server log data

```
cd home/sanjay/apache-flume-1.6.0-bin
apache-flume-1.6.0-bin/conf/nano flume.conf
```

```
python generate_logs.py
```

When the log file has been generated, a timestamp will appear, and the command prompt will return to normal. It may take several seconds to generate the log file. Sample data will be as shown below:



```
21.115.155.111 - - [01/Jan/2012:12:07:48 +0530] "GET /digital-cameras/digital-camera/sony-ex-dsc-qx100-point-shoot-digital-camera-black.html HTTP/1.1"
200 1478 "Mozilla/5.0 (Windows; U; Windows NT 6.1; en US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17" "-"
166.42.128.252 - - [01/Jan/2012:12:09:30 +0530] "GET /digital-cameras/digital-camera/canon-powershot-sx50-hs-point-shoot-camera.html HTTP/1.1" 200 195
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17" "-"
196.34.35.201 - - [01/Jan/2012:12:18:18 +0530] "GET /tvs-audio/blu-ray-dvd-players/d-m-holdings-inc-denor-dbt-1713ud-blu-ray-player.html HTTP/1.1" 200
1505 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
91.238.209.0 - - [01/Jan/2012:12:28:04 +0530] "GET /home-appliances/fans/reconnet-rhcfg-1201-ceiling-fan.html HTTP/1.1" 200 773 "Mozilla/5.0 (Windows NT
6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
114.131.104.220 - - [01/Jan/2012:12:44:23 +0530] "GET /catalogsearch/result/index/?dir=asc&order=price&q=samsung HTTP/1.1" 200 1168 "AdsBot-Google-
Mobile (http://www.google.com/mobile/adsbot.html) Mozilla (iPhone U CPU iPhone OS 3 0 like Mac OS X) AppleWebKit (KHTML, like Gecko) Mobile Safari" "-"
15.196.61.220 - - [01/Jan/2012:12:53:03 +0530] "GET /home-appliances/fans/reconnect-rhtfg-5501-table-fan.html HTTP/1.1" 200 1355 "Mozilla/5.0
(compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"
119.170.181.99 - - [01/Jan/2012:01:05:06 +0530] "GET /digital-cameras/dslr-cameras/canon-eos-7d-dslr-camera-black-with-ef-s18-135is-lens.html HTTP/1.1"
200 803 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
41.28.154.14 - - [01/Jan/2012:01:19:35 +0530] "GET /tvs-audio/flat-television/samsung-ua22f5100ar-led-tv-22-inch-56-cm.html HTTP/1.1" 200 1297 "AdsBot-
Google Mobile (http://www.google.com/mobile/adsbot.html) Mozilla (iPhone U CPU iPhone OS 3 0 like Mac OS X) AppleWebKit (KHTML, like Gecko) Mobile
Safari" "-"
168.105.153.185 - - [01/Jan/2012:01:24:23 +0530] "GET /tvs-audio/mp3-players-ipods/apple-ipod-touch-md718hn-a-mp3-player-blue.html HTTP/1.1" 200 810
"Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
85.36.161.194 - - [01/Jan/2012:01:26:05 +0530] "GET /home-appliances/fans/usha-mist-air-ex-standard-pedestal-fan.html HTTP/1.1" 200 118 "Mozilla/5.0
(compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"
225.171.105.168 - - [01/Jan/2012:01:38:48 +0530] "GET /home-appliances/fans/reconnet-rhcfg-1201-ceiling-fan.html HTTP/1.1" 200 1830 "Mozilla/5.0
(Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17" "-"
90.160.130.234 - - [01/Jan/2012:01:52:06 +0530] "GET /tvs-audio/speciality-speakers/jbl-pulse-speaker-black.html HTTP/1.1" 200 1172 "Mozilla/5.0
(Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
133.128.48.53 - - [01/Jan/2012:01:55:42 +0530] "GET /mobiles/smart-phones/sony-xperia-m-dual-android-smart-phone-white.html HTTP/1.1" 200 1466
"Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17" "-"
97.134.59.178 - - [01/Jan/2012:02:00:02 +0530] "GET /tvs-audio/hifi-system/sony-gtk-n1bt-3-channels-mini-hi-fi-system.html HTTP/1.1" 200 1788 "AdsBot-
Google-Mobile (http://www.google.com/mobile/adsbot.html) Mozilla (iPhone U CPU iPhone OS 3 0 like Mac OS X) AppleWebKit (KHTML, like Gecko) Mobile
Safari" "-"
```

Fig 9: Snapshot of sample weblog data

2. Creating tables from weblog data

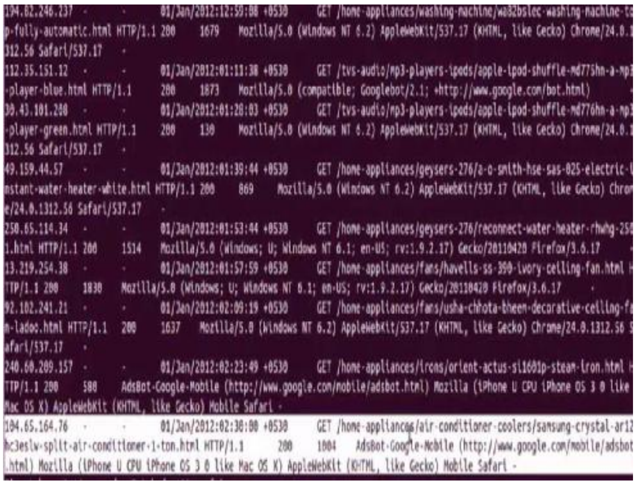
```
CREATE EXTERNAL TABLE weblog_data
(
remoteIP STRING,
remoteLogname STRING,
user STRING,
time STRING,
request STRING,
statusCode STRING,
byteString STRING,
referral STRING,
browser STRING
)
ROW FORMAT SERDE
'org.apache.hadoop.hive.contrib.serde2.R
egexSerDe'
WITH SERDEPROPERTIES (
"input.regex" = '^(\\S+) (\\S+) (\\S+)
\\[(.+?)\\] \"([^\"]*)\" (\\S+) (\\S+)
\"([^\"]*)\" \"([^\"]*)\"',
"output.format.string" = "%1$s %2$s %3$s
%4$s %5$s %6$s %7$s %8$s %9$s")
LOCATION '/user/sanjay/weblogs';
```

Now we will start hive and use following command to create external table:

Now we can see our data from the table using following hive command

Now data will be in the desired format as you can see below:

```
hive>select * from weblog_data;
```



```
194.82.246.237  -       -       01/Jan/2012:12:59:08 +0530      GET /home-appliances/washing-machine/wa82bslec-washing-machine-to
p-fully-automatic.html HTTP/1.1 200     1679    Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1
312.56 Safari/537.17    -
112.35.151.12   -       -       01/Jan/2012:01:11:38 +0530      GET /tvs-audio/mp3-players-ipods/apple-ipod-shuffle-md775hn-a-mp3
-player-blue.html HTTP/1.1      200     1873    Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)        -
39.43.101.200   -       -       01/Jan/2012:01:28:03 +0530      GET /tvs-audio/mp3-players-ipods/apple-ipod-shuffle-md776hn-a-mp3
-player-green.html HTTP/1.1     200     130     Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1
312.56 Safari/537.17    -
49.159.44.57    -       -       01/Jan/2012:01:39:44 +0530      GET /home-appliances/geysers-276/a-o-smith-hse-sas-025-electric-i
nstant-water-heater-white.html HTTP/1.1 200     869     Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrom
e/24.0.1312.56 Safari/537.17    -
250.65.114.34   -       -       01/Jan/2012:01:53:44 +0530      GET /home-appliances/geysers-276/reconnect-water-heater-rhwg-250
1.html HTTP/1.1 200     1514    Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17      -
13.219.254.38   -       -       01/Jan/2012:01:57:59 +0530      GET /home-appliances/fans/havells-ss-390-ivory-ceiling-fan.html H
TTP/1.1 200     1830    Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17      -
92.102.241.21   -       -       01/Jan/2012:02:09:19 +0530      GET /home-appliances/fans/usha-chhota-bheem-decorative-ceiling-fa
n-ladoo.html HTTP/1.1   200     1637    Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56 S
afari/537.17    -
240.60.209.157  -       -       01/Jan/2012:02:23:49 +0530      GET /home-appliances/irons/orient-actus-si1601p-steam-iron.html H
TTP/1.1 200     580     AdsBot-Google-Mobile (http://www.google.com/mobile/adsbot.html) Mozilla (iPhone U CPU iPhone OS 3 0 like
Mac OS X) AppleWebKit (KHTML, like Gecko) Mobile Safari -
104.65.164.76   -       -       01/Jan/2012:02:30:00 +0530      GET /home-appliances/air-conditioner-coolers/samsung-crystal-ar12
hc3eslw-split-air-conditioner-1-ton.html HTTP/1.1       200     1004    AdsBot-Google-Mobile (http://www.google.com/mobile/adsbot
.html) Mozilla (iPhone U CPU iPhone OS 3 0 like Mac OS X) AppleWebKit (KHTML, like Gecko) Mobile Safari -
```

Fig 9: Snapshot of parsed output data

Similarly we can use different hive commands for analysis which could be predictive, prescriptive, diagnostic and descriptive.

Conclusion:

It was an effort to emphasizing the use of new technology in today's business, in today's world social media is drastically changing the way of communication, and peoples are habituated to express his opinion, views on the social media.By using the Big Data technology we can utilize these social media data for the analysis purpose, which can be linked with the business to know the feedback of the users and people from that zone about a particular product,because big data will fetch the data from the server of that zone. Still variety of analysis can be done by using Hadoop ecosystem.

Reference:

[1] Hadoop: The Definitive Guide by tom white, O'REILLY publication.

[2] Apache Cloudera- http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht_example_4_sentiment_analysis.html

[3] http://hortonworks.com/hadoop/data-science/

[4] http://hortonworks.com/hadoop-tutorial/how-to-refine-and-visualize-server-log-data/