12/12/2014

Twitter Big Data Analytics Using Hive

CS5590PB Project Report

**Project Members:**

**Srikar Reddy Mallareddygari (sm8xd)**

**Gouri Priya Vangavargu (gvybf)**

**Pavan Kumar Bollaram (pbc6d)**

**Lavanya Kumar Somu (lsxp7)**

**Motivation**

Mobile phones becomes an essential companion in our day to day lives. They helps us to keep in touch with friends, family, colleagues, access email, browse internet etc. Mobile phones were brought to life with the help of an operating system. In the present world, Android and IOS are having the major mobile operating systems market share in the world.

Android holds a market share of 61.9% in the current US market while Apple`s IOS holds only 32.5% of share. Internationally android`s market share is even lot better when compared to Apple`s IOS. So keeping these facts in mind we are inspired to perform big data analytics on tweets related to android operating system.

**Introduction**

Big data is the buzzing word in the present software industry. Huge amounts of data is being generated daily from various sources. Companies are trying to perform analytics on big data and get some valuable output which gives an edge over there competitors. In order to achieve this we need to program map reduce jobs in Hadoop ecosystem. It is very difficult to develop the code and reuse it for different business cases. On the other hand, People are very much comfortable to query data using SQL like queries.

A team of developers at Facebook developed a dataware house tool namely called as HIVE. Hive supports the queries like SQL type which is called as HiveQL. These queries are compiled as map reduce jobs and are executed using Hadoop. Through HiveQL we can plugin custom map reduce scripts into the queries.It is easy to extract, transform and load the data using Hive NoSQL. In Hive Query execution is done using Map Reduce.

**System requirements**

* Environment: IBM Info Sphere Big Insights v3.0
* NoSQL Tool: Hive
* Visualization: IBM Big Sheets.
* Programming Languages: Python, Java
* Data Source : Twitter Data on keyword “android”
* Volume of Data : 783,876 Tweets
* Data Size : 3.87 GB
* Data Format : json

**Design Architecture**

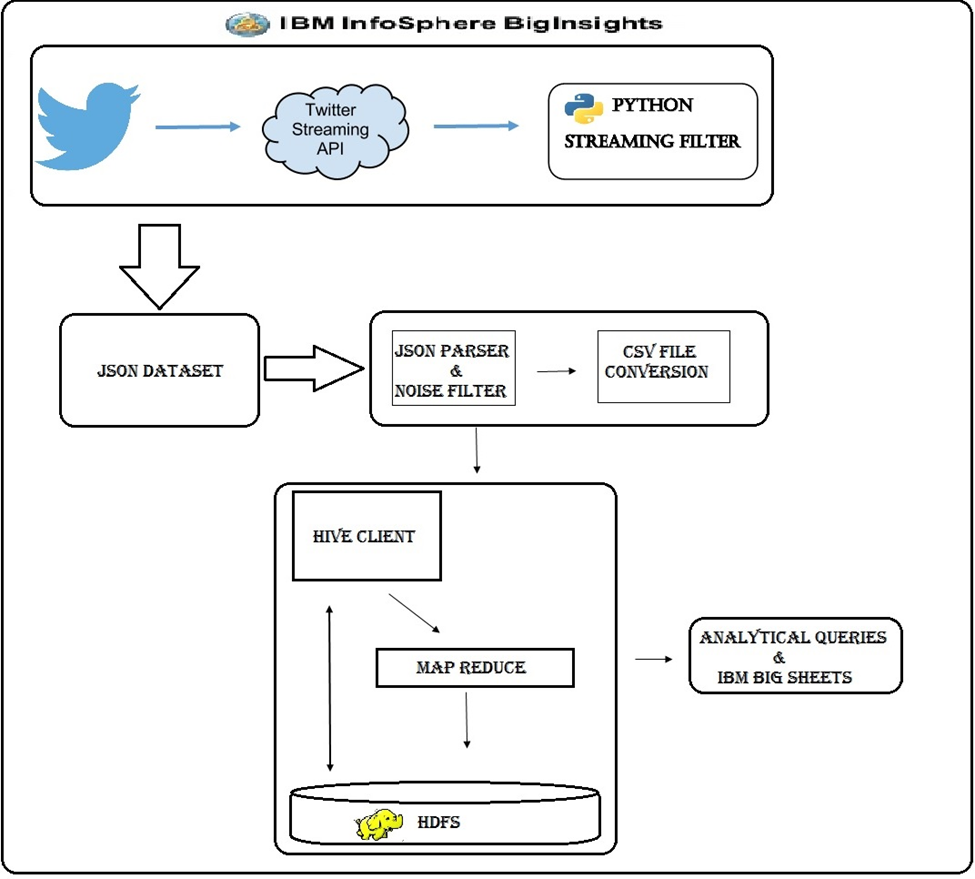


Fig 1: System Architecture

**Implementation**

**Approach:**

The main idea of the project is to collect tweets using a keyword “android” from a twitter streaming API and to analyze the data.

**Implementation steps:**

**Note:** All the below stages were implemented in IBM Big platform

Stage 1:

* A twitter application is created to access the streaming tweets from twitter website.
* Required authorizations were created for connecting to the API’s from python code.
* A library named tweepy from python is used to download streaming tweets from twitter with a streaming filter i.e., ‘android’.
* All the streaming tweets were collected into a json file to the local machine.

GitHub URL: <https://github.com/pavankumar-b/PB/tree/master/stage1_Tweetdownload>

Stage 2:

* A JSON2CSV.java class file is developed to filter the noise and convert the json file to csv format.

GitHub URL: <https://github.com/pavankumar-b/PB/tree/master/stage2_Json2Csv/json2csv>

Stage 3:

* A Hive table named **master\_tweets** is created using Row Format as csv serde.
* All the Tweets were loaded into the above hive table.
* Again, three tables named **tweet\_info, user\_info, user\_count\_info** were created to store the individual data from master table.
* Analytical queries were executed in hive shell to view and analyze the data.

GitHub URL: <https://github.com/pavankumar-b/PB/tree/master/stage3_TerminalCommands>

Stage 4:

* All the Analytical queries output is stored in external hive tables.
* External table data is copied from Hadoop file system to local system using hadoop fs –copyToLocal command.
* We used IBM Big Sheets for visualization of the results. Once the results are available in local, We uploaded that results in to the Infosphere files tab.
* In the tool we convert the comma separated text format data in to sheet format. Then we select the line reader as comma separated value data.
* We save this as a master workbook then the control is redirected to bigsheets tab. Here we select the required visualization for our results data.

**Tables Information**

**Table 1:**

**Syntax:**

create table master\_tweets(

created\_at string,

id bigint,

id\_str string,

text string,

user\_id bigint,

user\_id\_str string,

user\_name string,

user\_screen\_name string,

user\_location string,

user\_protected boolean,

user\_verified boolean,

user\_followers\_count bigint,

user\_friends\_count bigint,

user\_listed\_count bigint,

user\_favourites\_count bigint,

user\_statuses\_count bigint,

user\_created\_at string,

user\_utc\_offset int,

user\_time\_zone string,

user\_geo\_enabled boolean,

user\_lang string,

user\_contributors\_enabled boolean,

user\_is\_translator boolean,

geo\_type string,

geo\_coord\_lat float,

geo\_coord\_long float,

coordy\_type string,

cordy\_coord\_lat float,

cordy\_coord\_long float,

place\_id string,

place\_type string,

place\_name string,

place\_ctry\_code string,

place\_country string,

place\_full\_name string,

retweet\_count int,

favorite\_count int,

favorited boolean,

retweeted boolean,

possibly\_sensitive boolean,

filter\_level string,

lang string,

timestamp\_ms string

)

ROW FORMAT serde 'com.bizo.hive.serde.csv.CSVSerde'

with serdeproperties(

"separatorChar" = "\,","quoteChar" = "\"") stored as textfile;

**Load Statement:**

load data local inpath "/home/biadmin/Desktop/PB2-2days/csv\_output/output1.csv" into table master\_tweets;

**Table 2: tweet\_info**

**Syntax:**

create table tweet\_info(

created\_at string,

id bigint,

id\_str string,

text string,

user\_id bigint,

geo\_type string,

geo\_coord\_lat float,

geo\_coord\_long float,

coordy\_type string,

cordy\_coord\_lat float,

cordy\_coord\_long float,

place\_id string,

place\_type string,

place\_name string,

place\_ctry\_code string,

place\_country string,

place\_full\_name string,

retweet\_count int,

favorite\_count int,

favorited boolean,

retweeted boolean,

possibly\_sensitive boolean,

filter\_level string,

lang string,

timestamp\_ms string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

stored as textfile;

**Load Statement:**

Insert overwrite table tweet\_info select distinct created\_at,id,id\_str,text,user\_id,geo\_type,geo\_coord\_lat,geo\_coord\_long,coordy\_type,cordy\_coord\_lat,cordy\_coord\_long,place\_id,place\_type,place\_name,place\_ctry\_code,place\_country,place\_full\_name,retweet\_count,favorite\_count,favorited,retweeted,possibly\_sensitive,filter\_level,lang,timestamp\_ms from master\_tweets;

**Table 3: user\_info**

**Syntax:**

create table user\_info(

user\_id bigint,

user\_id\_str string,

user\_name string,

user\_screen\_name string,

user\_location string,

user\_protected boolean,

user\_verified boolean,

user\_created\_at string,

user\_utc\_offset int,

user\_time\_zone string,

user\_geo\_enabled boolean,

user\_lang string,

user\_contributors\_enabled boolean,

user\_is\_translator boolean

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

stored as textfile;

**Load Statement:**

Insert overwrite table user\_info select distinct user\_id,user\_id\_str,user\_name,user\_screen\_name,user\_location,user\_protected,user\_verified,user\_created\_at,user\_utc\_offset,user\_time\_zone,user\_geo\_enabled,user\_lang,user\_contributors\_enabled,user\_is\_translator from master\_tweets;

**Table 4:** user\_count\_info

**Syntax:**

create table user\_count\_info(

user\_id bigint,

user\_followers\_count bigint,

user\_friends\_count bigint,

user\_listed\_count bigint,

user\_favourites\_count bigint,

user\_statuses\_count bigint

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

stored as textfile;

**Load Statement:**

Insert overwrite table user\_count\_info select user\_id,max(user\_followers\_count),max(user\_friends\_count),max(user\_listed\_count),max(user\_favourites\_count),max(user\_statuses\_count) from master\_tweets group by user\_id;

**Analytic Queries and Visualization**

**Query 1:**

select count(ti.text), ui.user\_id, ui.user\_name, ui.user\_screen\_name From tweet\_info ti JOIN user\_info ui ON (ti.user\_id = ui.user\_id) GROUP BY ui.user\_id, ui.user\_name, ui.user\_screen\_name;

**Description 1: To find the users contribution towards tweets in dataset.**

**Visualization 1:**

****

Fig 2: Tweet\_Contribution\_WordCloud

**Query 2:**

select ui.user\_id,ui.user\_name,ui.user\_screen\_name,uci.user\_statuses\_count from user\_info ui JOIN user\_count\_info uci ON (ui.user\_id=uci.user\_id) ORDER BY uci.user\_statuses\_count DESC;

**Description 2: To identify Active Users in twitter from the given users dataset.**

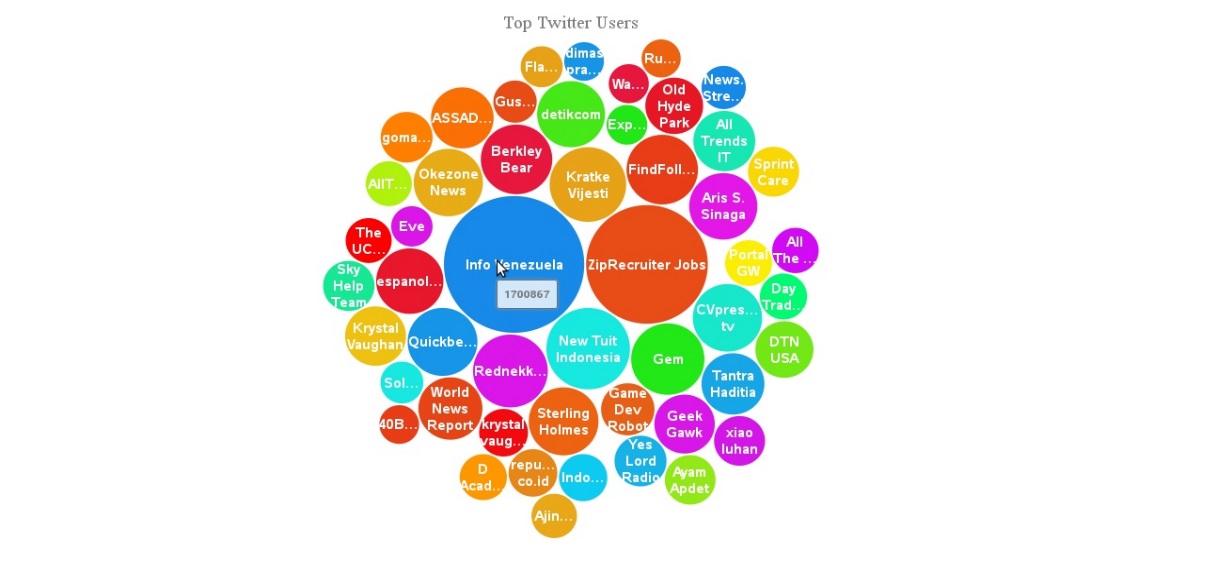
**Visualization 2:**

Fig 3: Active\_Userschart\_TagCloud

**Query 3:**

select text,geo\_coord\_lat,geo\_coord\_long from tweet\_info where geo\_coord\_lat is not null and geo\_coord\_long is not null;

**Description 3: To plot the geographical locations of tweets from dataset.**

**Visualization 3: TBD**

Fig 4: Geoplot\_Tweet\_Map

**Query 4:**

select ui.user\_name,ui.user\_screen\_name,uci.user\_followers\_count from user\_info ui JOIN user\_count\_info uci ON(ui.user\_id=uci.user\_id) ORDER BY uci.user\_followers\_count DESC;

**Description 4: To list out the popular users from the given dataset.**

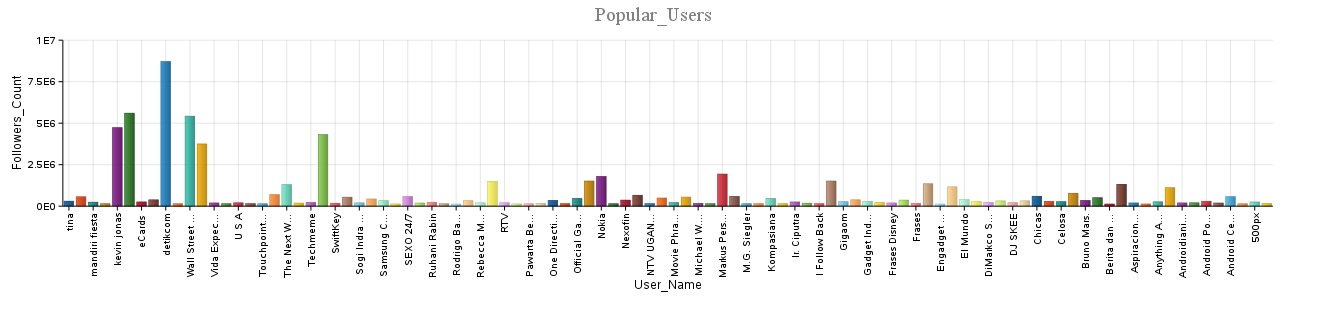
**Visualization 4:**

Fig 5: Popular\_Users\_Bargraph

**Query 5:**

select place\_country country,count(\*) count from tweet\_info group by place\_country;

**Description 5: To find the tweet contribution from various countries.**

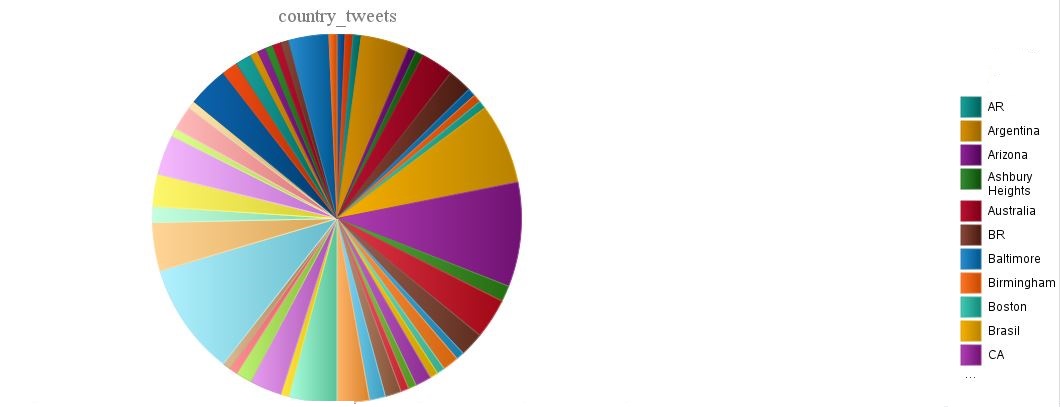
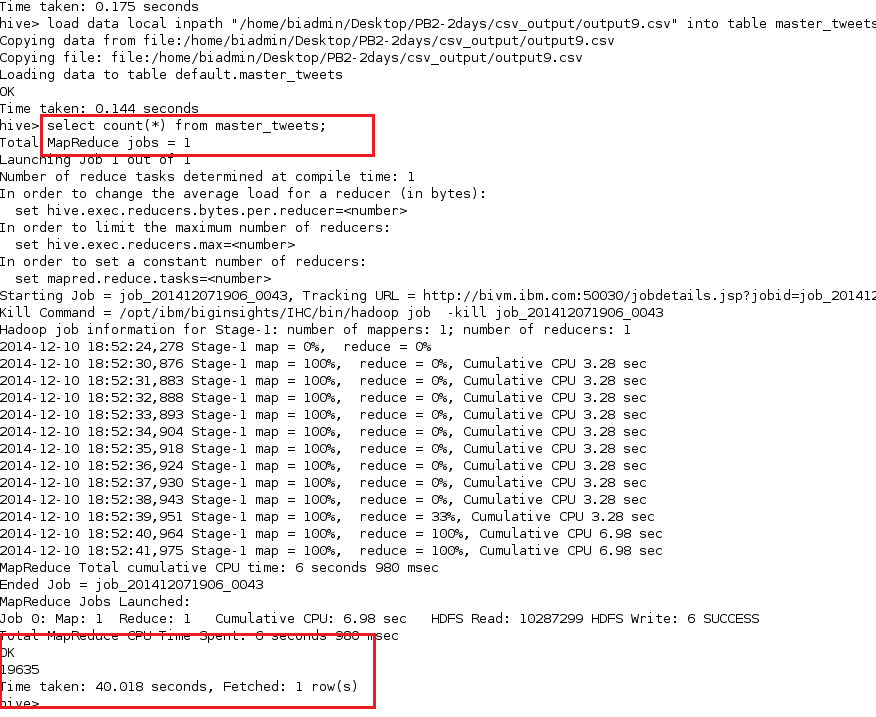
**Visualization 5:**

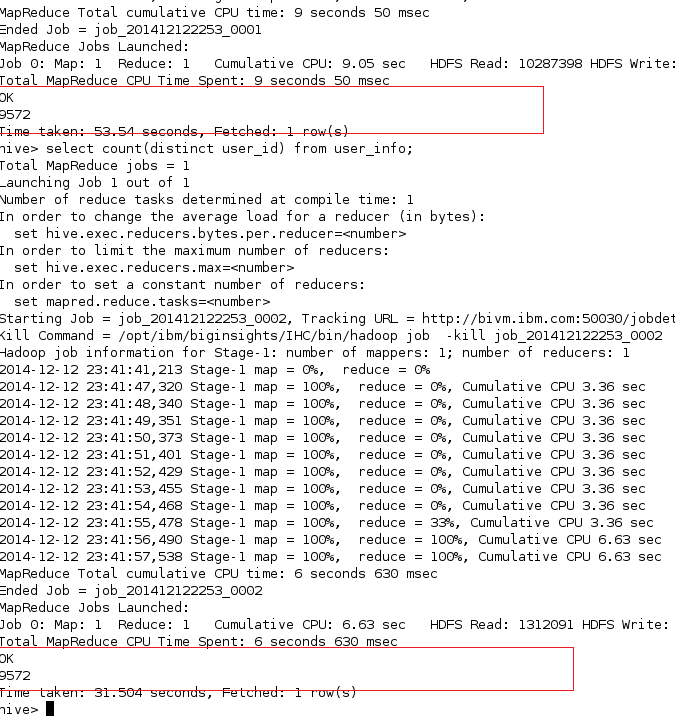
Fig 6: Country\_Tweets\_PieChart

**Testing Details**

1. Twitter Streaming Data from Python contains a lot of noise data in the tweets.
   1. Around 3.87 Giga Bytes of Data of 783876 tweets were collected but there is a noise data and no useful information in most of the tweets.
2. JSON Parser: Before converting the json file to csv file most of the tweets were filtered to get useful information.
   1. Around 19635 tweets were converted to csv file for hive tables.
3. A CSV File with 19635 is loaded to hive tables. All the data is loaded into master tables.



1. From Master table master\_tweets data is inserted into 3 other tables for extracting the individual records data.
   1. Data from master tables and 3 other tables were matched with the results count.



1. Our query 2 is useful to identify Active Users in twitter from the given users dataset which means the twitter page having highest number of tweets tweeted. This can be observed in the below two figures. In fig 3 we can see 1700867 tweets by Info Venezuela twitter page, we observed the tweet count in Info Venezuela(@Info\_Ve) page as 1708271. This is a clear indication of successful execution of our joins queries.

**** Fig 7a: Query 2 Graphical output. Fig 7b: Tweet count in official page.

**Queries to be visualized:**

1:  
-- Language based tweets count from dataset

select lang, count(\*) from tweet\_info group by lang having lang is not null;

2:  
-- Total no of Retweeted tweets in a dataset

select count(\*) from tweet\_info where text like "RT%";

3:  
-- Top Friends Count from a given dataset

select ui.user\_name,ui.user\_screen\_name,uci.user\_friends\_count  
from user\_info ui JOIN user\_count\_info uci  
ON(ui.user\_id=uci.user\_id)  
ORDER BY uci.user\_friends\_count DESC;

**References:**

<http://bgr.com/2014/07/01/android-market-share-2014/>

<http://infolab.stanford.edu/~ragho/hive-icde2010.pdf>

<https://hive.apache.org/>