Project Phase2

Report

BY

Raviteja Reddy Kandakatla

Sandeep Siddhi

Sowmya Gangidi

Contents:

Introduction

Big data refers to extremely huge data sets that have grown beyond the ability to analyze using traditional data processing tools. It refers to both structured and unstructured data. The capability of storing such large sets of data isn't new. What's new is the ability to analyze this data quickly and effectively. Our project emphasizes on analyzing (i.e., visualizing using the charting libraries) such large volumes of data collected from twitter.

Specific requirements

Environment: Ubuntu

Framework to collect tweets: .NET (C#.NET)

Data format: JSON

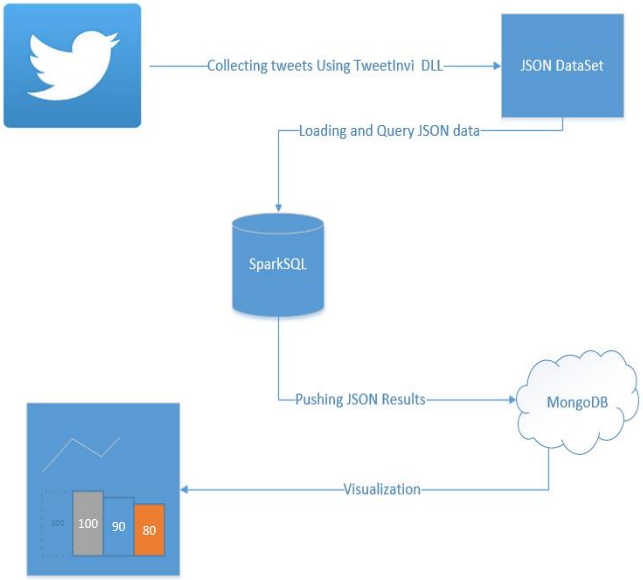
SQL tool used: Spark SQL

DB used to store results for visualization: MongoLab(API services)

Visualization: Plotly, MorrisJS

Client side Scripting: Angular JS

Architecture:



Implementation

Step 1: Tweet collection

Tweets are collected using Tweetinvi C# library. The tweets are saved in the JSON format.

Step 2: Save the JSON file into a temporary table

Spark SQL provides SQL Context as sqlContext. The JSON data is stored into a temporary table.

```
val jsonData = sqlContext.jsonFile("filepath")
```

```
jsonData.registerTempTable("TableName")
```

Step 3: Querying the Tweet File

SparkSQL is used to query the tweets in temporary table.

```
val resultset = sqlContext.sql("<<query>>")
```

Step 4: Saving the query results

The results of the query are stored in an output file using the command

```
 resultset.write.json("outputfilepath")
```

Step 5: Pushing the results to the MongoLab

```
mongoimport -h ds057934.mongolab.com:57934 -d pbproject -c <collection> -u <user> -p <password> -
-file <input file>
```

The output file is stored is imported to the mongo lab

Step 6: Fetching results as a service

Using AngularJS scripting the results are fetched as a service in JSON format

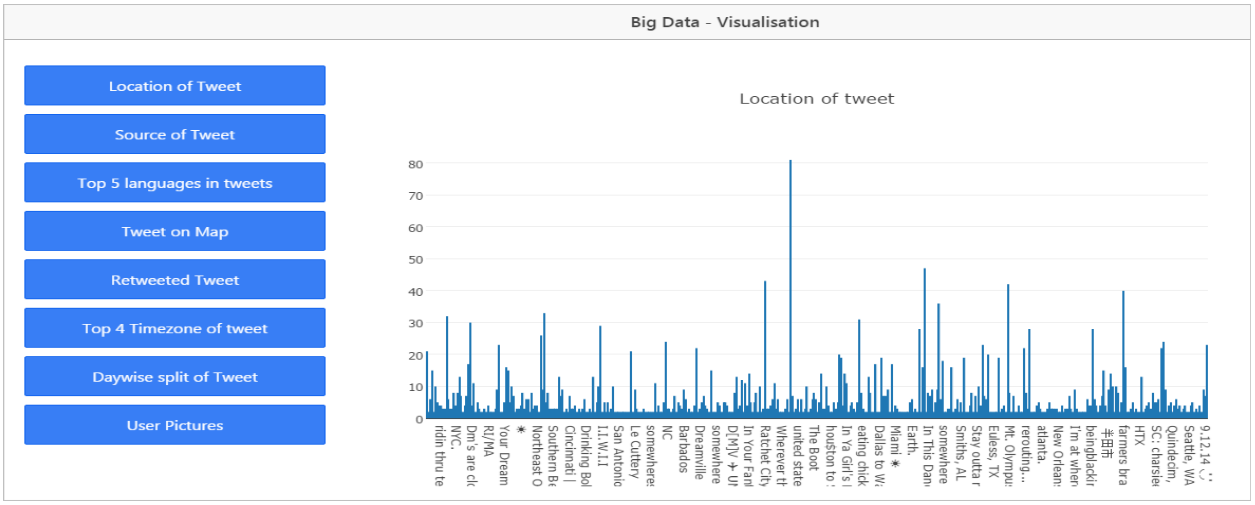
Analytic Queries with Visualization:

Query 1

Select user.location as location, count (user.location) as locationCount from LocationTable group by user.location

Query Description:

The above query results the total tweet count generated from a specific location in a dataset.

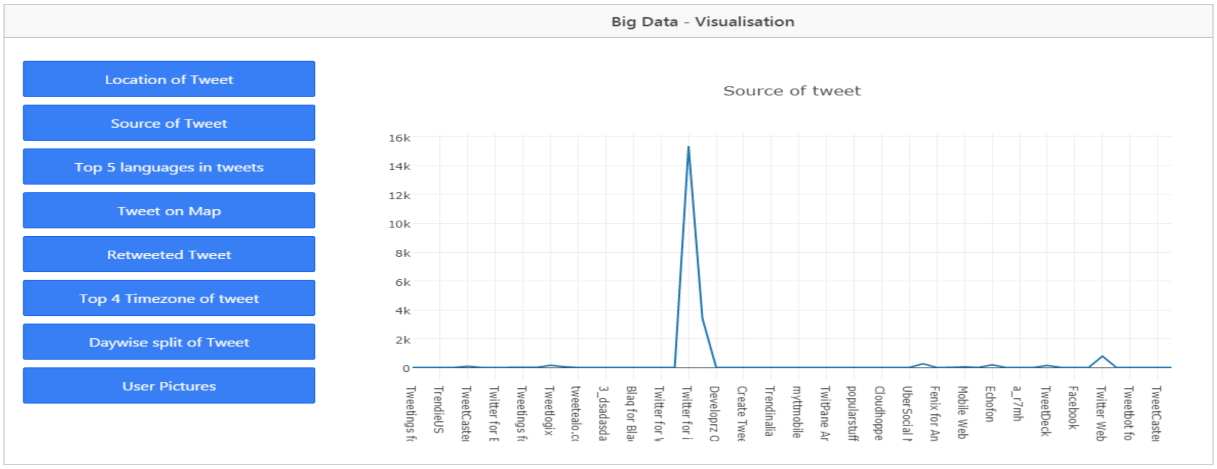


Query 2

Select source,count(source) from sourcetable group by source

Query Description:

The above query results the total count of sources from which the tweets are being tweeted in the dataset.

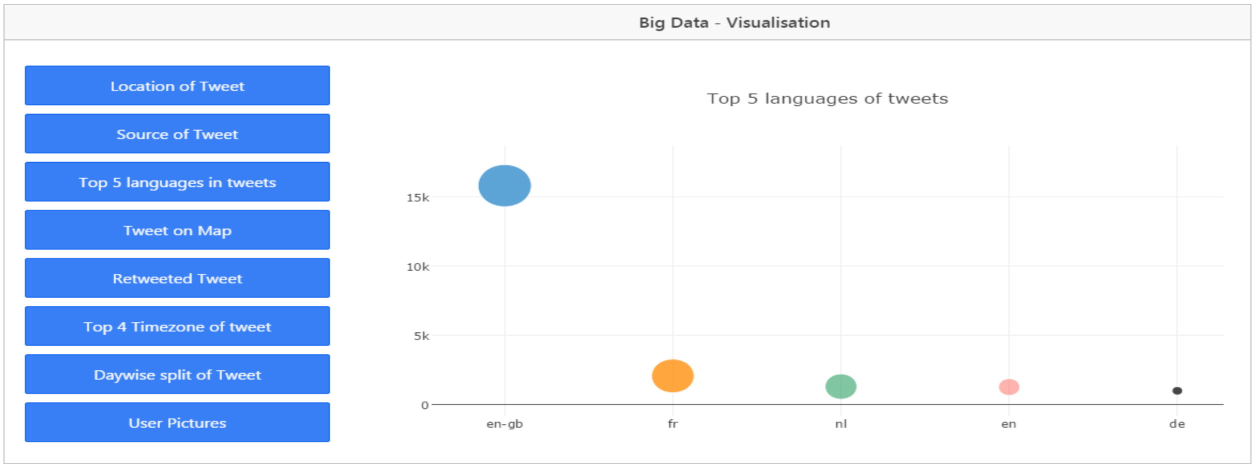


Query 3

Select user.lang,count(user.lang) from workTable group by user.lang order by count(user.lang) desc limit 5

Query Description:

The above query results the count of top 5 languages the users are tweeting from a dataset.

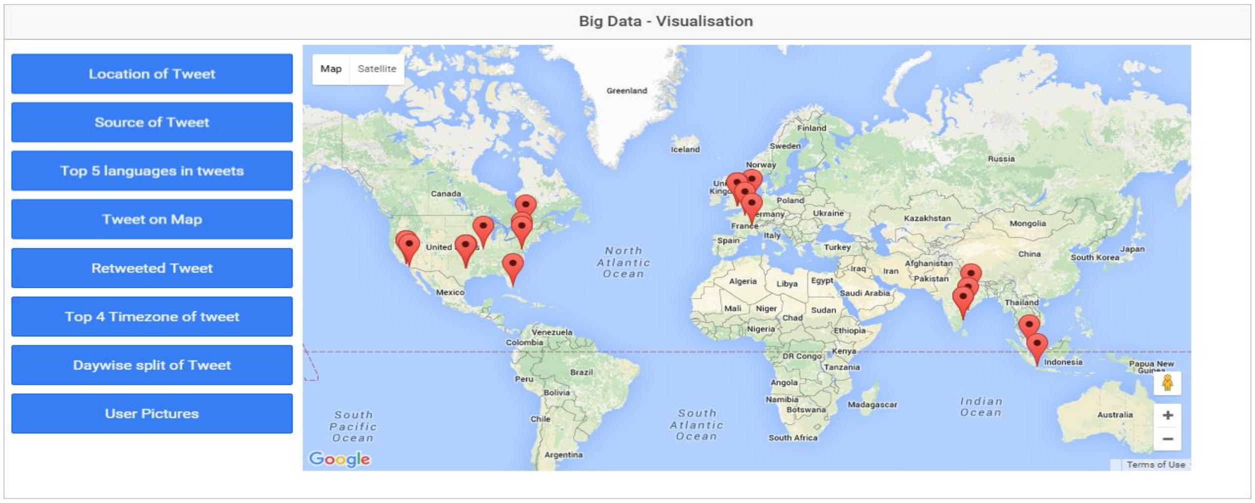


Query 4

Select text, geo.coordinates from geotable where geo.coordinates is not null"

Query Description:

The above query results the coordinates and the text of the user from a dataset if the location sharing is on.



Query 5

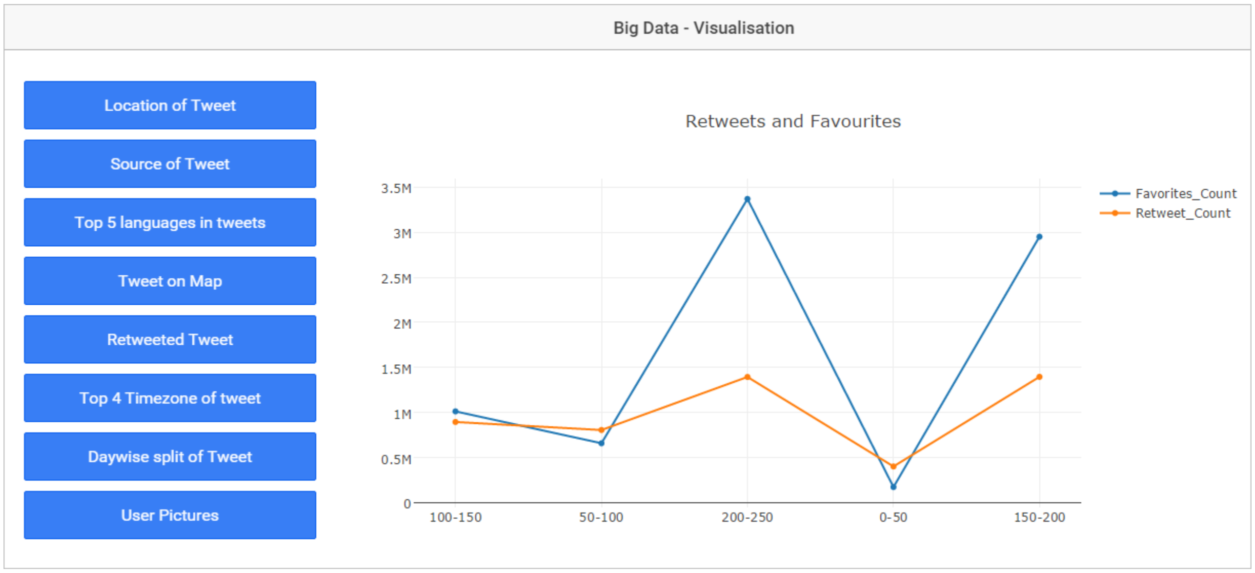
Select '0-50' as followers_range,sum(user.favourites_count) as favourites,sum(retweet_count) as retweet_count from logs where user.followers_count > 0 and user.followers_count < 50

Query Description:

The above query results the sum of favourites count and retweets count for the people who have their followers range between 0 and 50.

In the above way we have also collected data for the ranges of 50-100,100-150,150-200 and 200-250.

The results are plotted in the below graph.

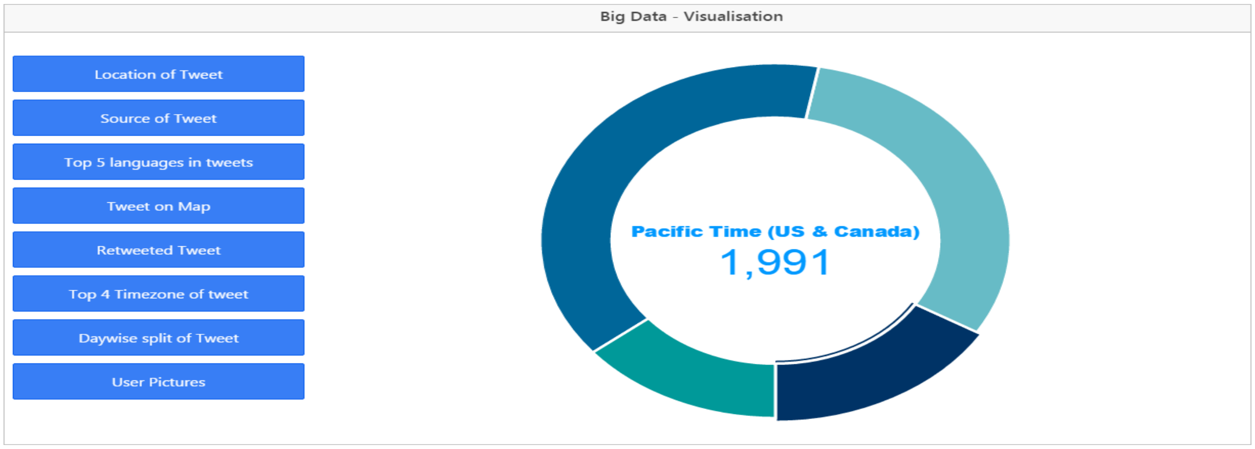


Query 6

Select user.time_zone,count(user.time_zone) from TimeZone group by user.time_zone order by count(user.time_zone) desc limit 4

Query Description:

The above query results the top 4 timezones from which the users are tweeting for a dataset.

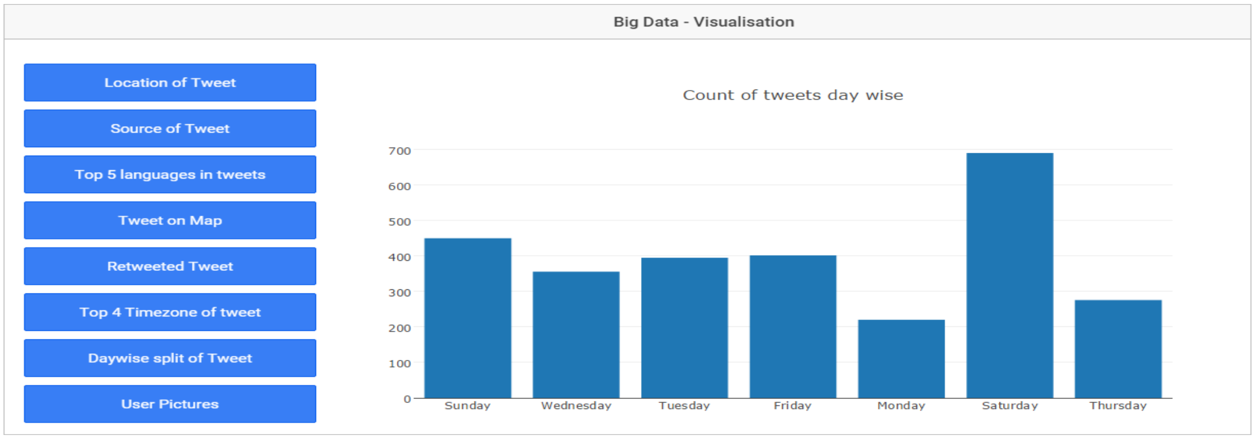


Query 7

Select 'Sunday' as Day, count(*) as count from timesnow where created_at like 'Sun%'

Query Description:

The above query results the total nuber of tweets recorded on a Sunday for a dataset.The results for the other days are also queried and plotted using the below graph.

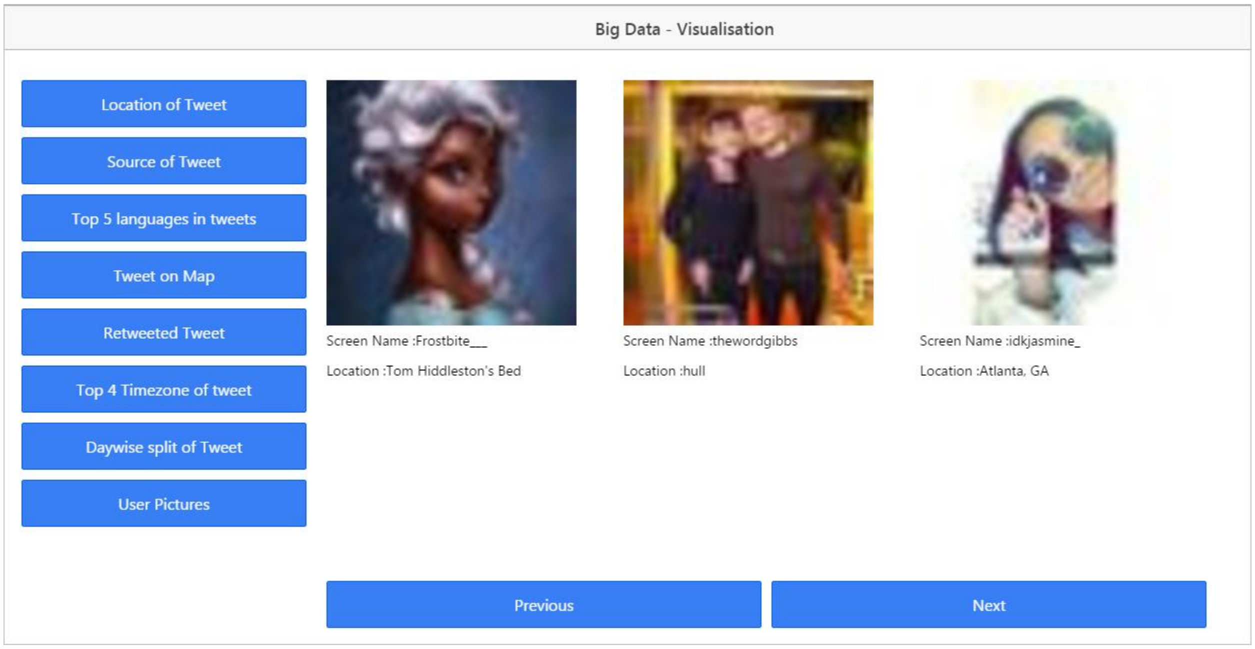


Query 8:

Select user.screen_name, user.profile_image_url_https,user.location from usertable where user.location is not null .

Query Description:

The above query results the screen name and the profile pic of the twitter user for a dataset.



Screen Name :Frostbite___

Location :Tom Hiddleston's Bed

Screen Name :thewordgibbs

Location :hull

Screen Name :idkjasmine_

Location :Atlanta, GA

References:

1. https://plot.ly/javascript/
2. http://morrisjs.github.io/morris.js/
3. https://mongolab.com/databases/pbproject#tools
4. https://www.mongodb.org/
5. http://spark.apache.org/sql/
6. https://tweetinvi.codeplex.com/