Twitter Miner

Contents

Introduction:

The goal of this project is to develop an application where users can enter a keyword and find distribution of tweets across United States. The twitter activity in various geographic locations will be shown to the user on an interactive map, where users can zoom in and out to interact and find out the percentage of tweets from each state. Users can also enter a keyword and get the sentiment associated with the keyword.

Motivation:

All public tweets posted on twitter are freely available through a set of APIs provided by Twitter. This tweet information can be used to find out trending topics online, popularity of a subject in any particular region, and perform statistical analysis. There are several platforms where user can enter a keyword and search for the tweets. We take the inspiration from www.trendsmap.com, where the global trending tweet tags are displayed on a map. But this webpage will not give the number of tweets or tweet density from different locations. We will design our application to show the density distribution of tweets on a map. We will also add additional feature (sentiment analysis) mentioned in the project description.

Technical Overview:

The project comprises of three core components namely:

1. Data Collection
2. Data Analysis
3. Data Visualization

We will elaborate each of the component in the following sections.

Data Collection:

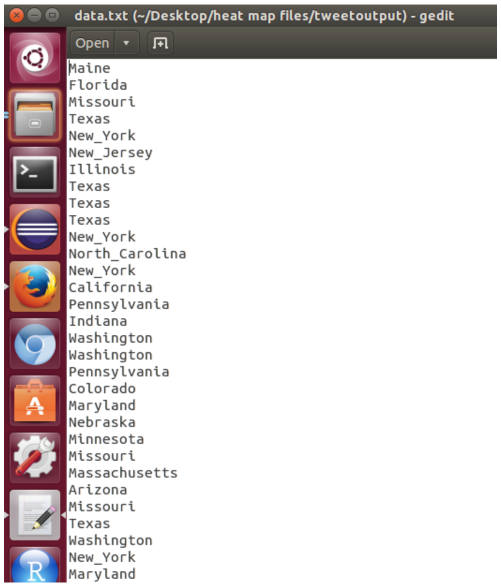
We used Twitter4j APIs to collect twitter data. The data can be fetched in two different ways using Twitter APIs: “Twitter Search” and “Twitter Stream”. Twitter search will search through existing tweets using Query class, whereas Twitter Stream will consume live tweets that match the keyword. Both APIs have restriction on the number of tweets that can be accessed. Using these APIs, we can access only 1% of total tweets. In Twitter Search API, the number of requests from one authorized user is limited to 15 requests per 15 minute interval. Making more than 15 requests will result in ErrorCode: 429. Fetching tweets using Twitter Search is a lot faster than fetching tweets using Twitter Stream. We didn't want to run Twitter APIs beforehand to collect data on selected topics, instead we wanted to design an application that can take any keyword and search real time to generate results. As real-time data fetching using Twitter Stream API will take considerable time, we decided to use Twitter Search API, which will result in lesser waiting time for the user to get results. The application can be scaled up to be more efficient by fetching all the tweets without restrictions by using Twitter Firehose API, which will allow unlimited access to the twitter information. Only partners of Twitter will have access to Twitter Firehose.

Heat Map:

Data Analysis:

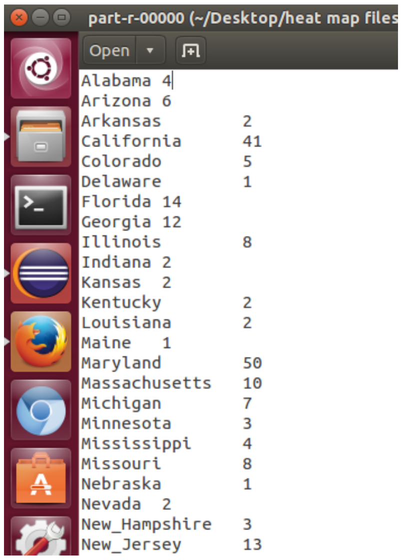
A heat map will show the density distribution of tweets on United States map. To achieve this functionality, we extracted location information from tweets. For implementing this functionality, we need only the location information. The location information coming from Twitter API is highly inconsistent, most of the times there will be null or some random string. As detecting the exact location from such inconsistent data is really difficult, we decided to limit our heat map to only United States. We found out that the tweets from United States mostly had the following location formats: (CityName, StateName) or (CityName, StateCode) or (StateName, USA) or (StateCode or USA). We designed a Java class that will take the location string as input, split it and verify if the location is in United States.

Once the predefined number of tweets are fetched, the list of states from where the tweets originated from is written to a text file. A sample output file is shown below:



```
Maine
Florida
Missouri
Texas
New_York
New_Jersey
Illinois
Texas
Texas
Texas
New_York
North_Carolina
New_York
California
Pennsylvania
Indiana
Washington
Washington
Pennsylvania
Colorado
Maryland
Nebraska
Minnesota
Missouri
Massachusetts
Arizona
Missouri
Texas
Washington
New_York
Maryland
```

This file is then passed to a Map Reduce job, which will count the number of times a state name has appeared in the document, which will indicate the number of tweets from that state. The Map Reduce output is shown below:



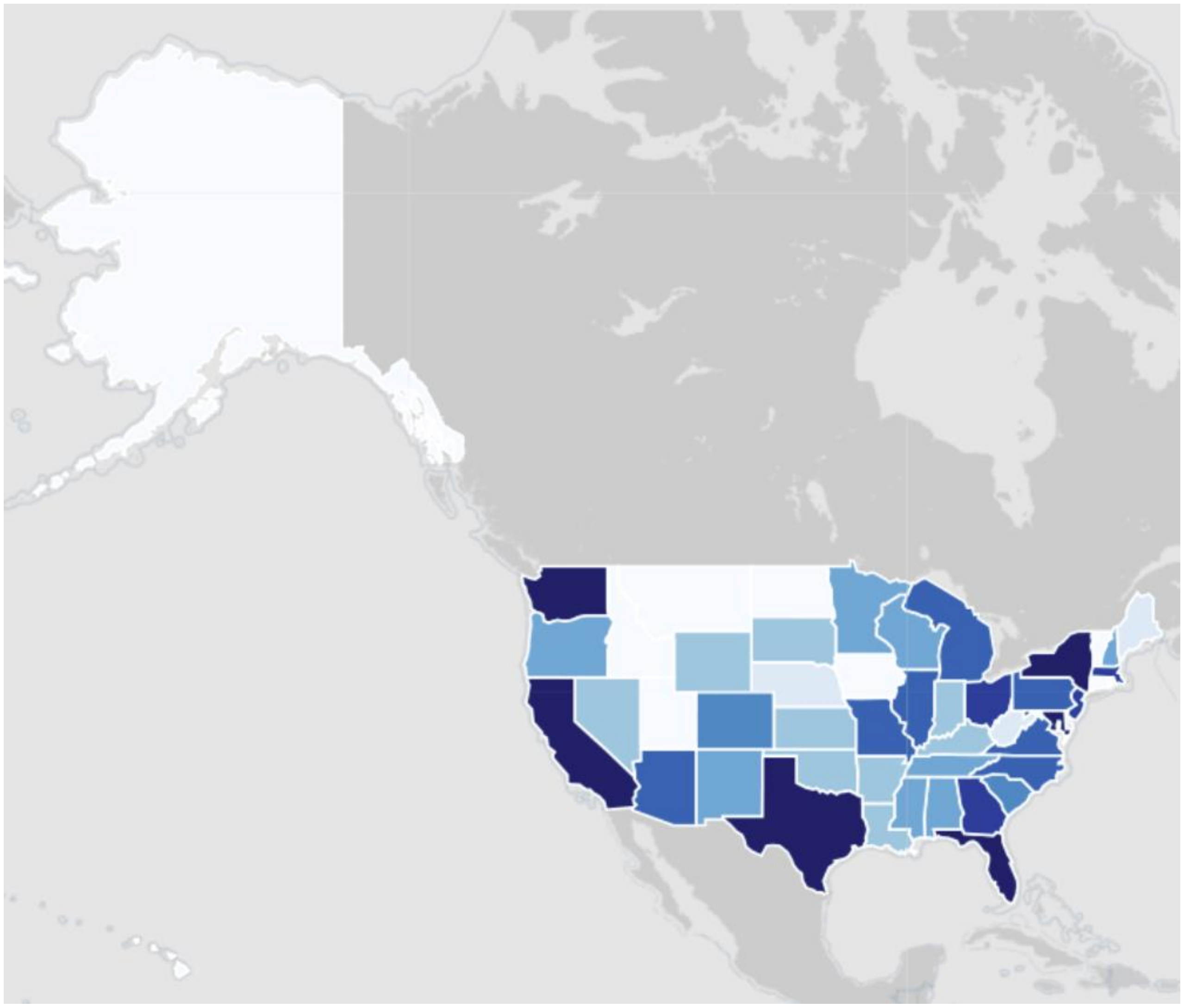
```
Alabama 4
Arizona 6
Arkansas        2
California      41
Colorado        5
Delaware        1
Florida 14
Georgia 12
Illinois        8
Indiana 2
Kansas  2
Kentucky        2
Louisiana       2
Maine   1
Maryland        50
Massachusetts   10
Michigan        7
Minnesota       3
Mississippi     4
Missouri        8
Nebraska        1
Nevada  2
New_Hampshire   3
New_Jersey      13
```

The Java application will read this output and calculate the percentage of tweets from each state. This information is finally showed on an interactive map.

Data Visualization:

We used Polymaps JavaScript APIs for displaying maps. Polymaps is a free JavaScript library for making dynamic, interactive maps in modern web browsers. Because Polymaps can load data at a full range of scales, it’s ideal for showing information from country level on down to states, cities, neighborhoods, and individual streets. The Polymaps API reads the tweet density information stored in variables in a Java Script. After reading the Map Reduce output, the application will calculate the percentage of tweets and dynamically writes the state names and corresponding percentages to a JavaScript file and the user is then redirected to the map page. The JavaScript in the HTML page will read the numerical data from the other JavaScript and will display the data.

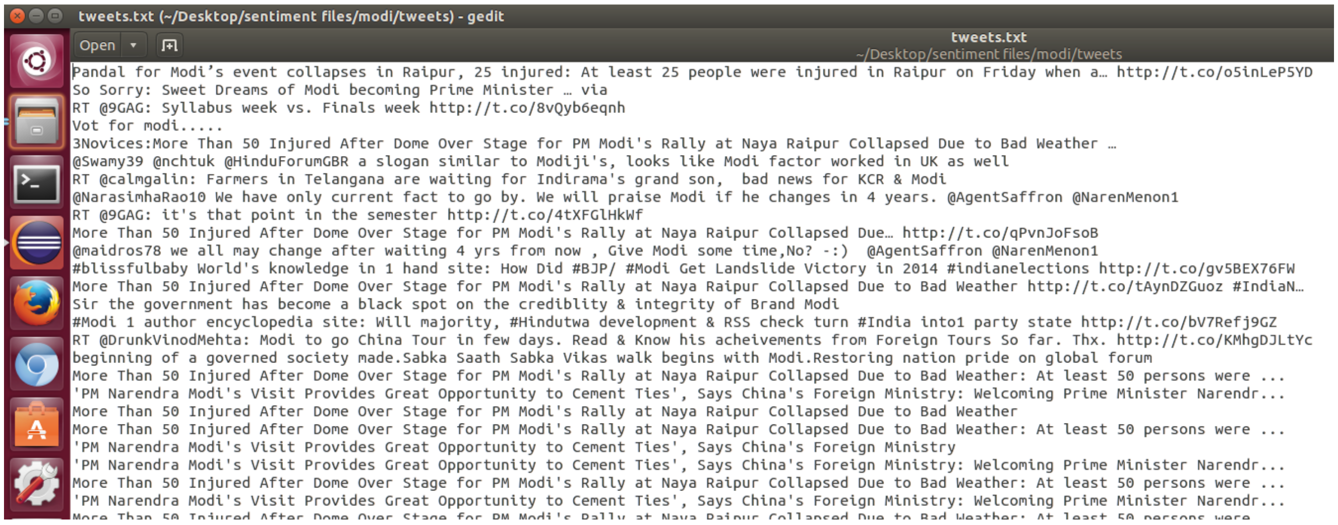
We used blue color scheme to display the density. Darker the color, higher the percentage of tweets from that state. If a state is colored white, that means no tweets originated from that state. Below is a screen shot of sample map:



Sentiment Analysis:

Data Analysis:

For sentiment analysis, we collected the tweets using the Twitter Search API and wrote them to a text file. A sample text file containing the tweets is shown below:


tweets.txt (~/Desktop/sentiment files/modi/tweets) - gedit
Open
tweets.txt
~/Desktop/sentiment files/modi/tweets

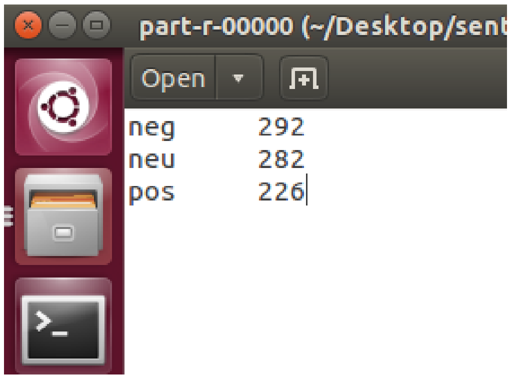
```
Pandal for Modi’s event collapses in Raipur, 25 injured: At least 25 people were injured in Raipur on Friday when a… http://t.co/o5inLeP5YD
So Sorry: Sweet Dreams of Modi becoming Prime Minister … via
RT @9GAG: Syllabus week vs. Finals week http://t.co/8vQyb6eqnh
Vot for modi.....
3Novices:More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather …
@Swamy39 @nchtuk @HinduForumGBR a slogan similar to Modiji's, looks like Modi factor worked in UK as well
RT @calmgalin: Farmers in Telangana are waiting for Indirama's grand son,  bad news for KCR & Modi
@NarasimhaRao10 We have only current fact to go by. We will praise Modi if he changes in 4 years. @AgentSaffron @NarenMenon1
RT @9GAG: it's that point in the semester http://t.co/4tXFGlHkWf
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due… http://t.co/qPvnJoFsoB
@maidros78 we all may change after waiting 4 yrs from now , Give Modi some time,No? -:)  @AgentSaffron @NarenMenon1
#blissfulbaby World's knowledge in 1 hand site: How Did #BJP/ #Modi Get Landslide Victory in 2014 #indianelections http://t.co/gv5BEX76FW
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather http://t.co/tAynDZGuoz #IndiaN…
Sir the government has become a black spot on the crediblity & integrity of Brand Modi
#Modi 1 author encyclopedia site: Will majority, #Hindutwa development & RSS check turn #India into1 party state http://t.co/bV7Refj9GZ
RT @DrunkVinodMehta: Modi to go China Tour in few days. Read & Know his acheivements from Foreign Tours So far. Thx. http://t.co/KMhgDJLtYc
beginning of a governed society made.Sabka Saath Sabka Vikas walk begins with Modi.Restoring nation pride on global forum
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather: At least 50 persons were ...
'PM Narendra Modi's Visit Provides Great Opportunity to Cement Ties', Says China's Foreign Ministry: Welcoming Prime Minister Narendr...
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather: At least 50 persons were ...
'PM Narendra Modi's Visit Provides Great Opportunity to Cement Ties', Says China's Foreign Ministry
'PM Narendra Modi's Visit Provides Great Opportunity to Cement Ties', Says China's Foreign Ministry: Welcoming Prime Minister Narendr...
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather: At least 50 persons were ...
'PM Narendra Modi's Visit Provides Great Opportunity to Cement Ties', Says China's Foreign Ministry: Welcoming Prime Minister Narendr...
More Than 50 Injured After Dome Over Stage for PM Modi's Rally at Naya Raipur Collapsed Due to Bad Weather: At least 50 persons were
```

Each line in the file represent one tweet. This file is given as input to another Map Reducer job (performs sentiment analysis) that will read each treat and assign a sentiment rating.

We tested both Stanford NLP and LingPipe APIs for performing sentiment analysis. While using Stanford NLP, we found that majority (80%) of tweets were given sentiment rating of “1”, which is not very informative. Hence we decided to use LingPipe API.

LingPipe is tool kit for processing text using computational linguistics. LingPipe's architecture is designed to be efficient, scalable, reusable, and robust. Using this API, we can train our model and generate a classifier which suits our needs. A classifier is a file which contains rules to assign sentiment rating for a tweet. For this project, we decided to use the default classifier provided with the APIs, which will be able to predict sentiment with 75% accuracy. The API will assign each tweet a sentiment rating of: Positive, Negative or Neutral.

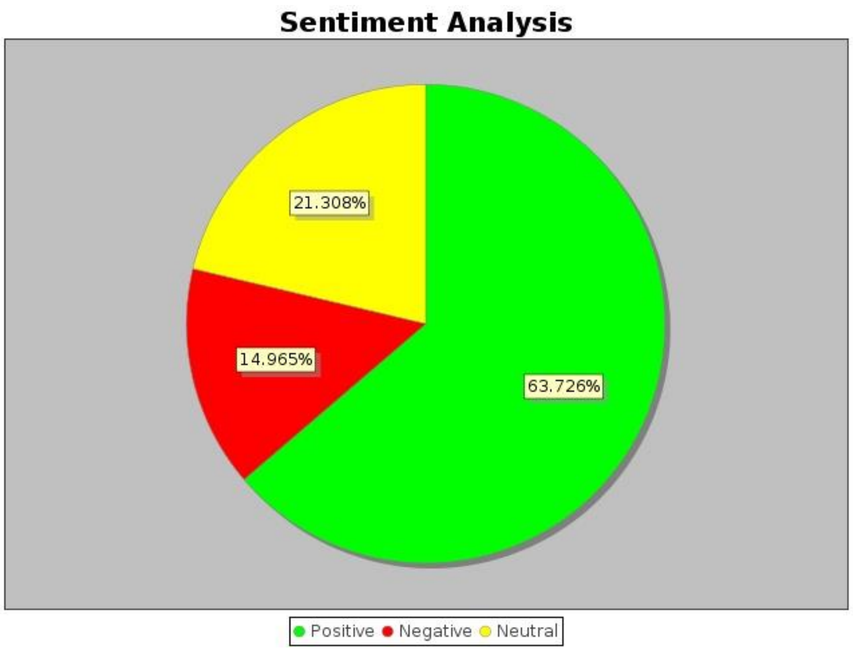
The Map Reduce job will give the total number of positive, negative and neutral tweets. Below is the screen shot of Map Reduce output:



```
neg	292
neu	282
pos	226
```

Data Visualization:

The Java application will read the above file and will generate a pie chart to display the percentages. We used JFreeChart library to generate pie chart. Below is a sample pie chart:



Code Explanation:

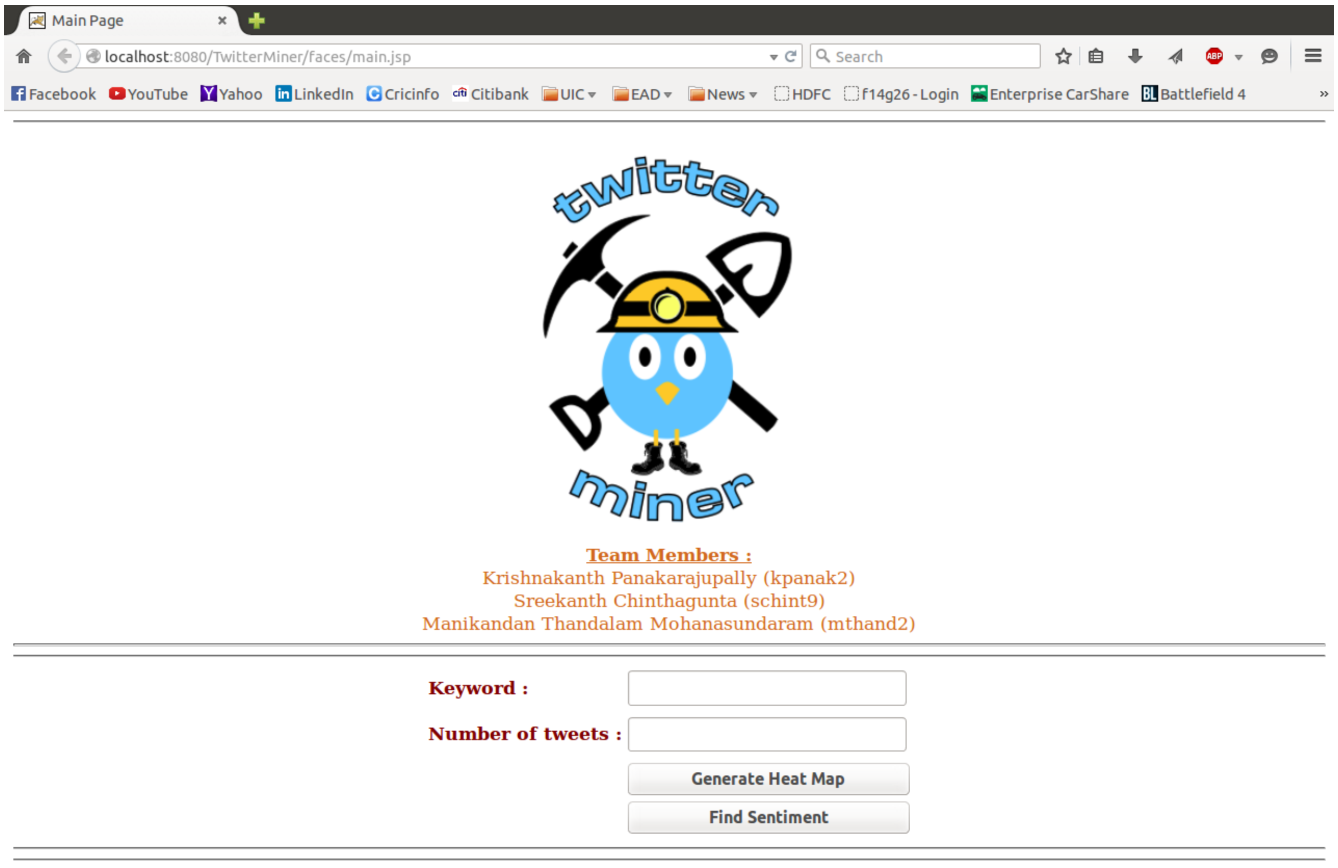
Package	Class	Description
com.pkg.twitter	Authorization	This class stores the twitter authorization information in private variables. These credentials are used across the application whenever there is a need to collect tweet information

	StreamingHeatMap	The methods in this class are executed when the user generates heat map. This class will fetch the location information from twitter and writes it to a file.
	StreamingSentiment Analysis	The methods in this class are called when user performs sentiment analysis. This class will search for tweets and writes them to a file.
	WriteFile	Utility class that is used to write location information to a file
	WriteToTweetsFile	Utility class that is used to write tweet information to a file
	ReadStates	This class reads a text file containing the list of state names and state codes and writes them into a static HashMap that will be used across the application. This class will also have a static string variable which stores the root path of the application after deploying on Tomcat server.
com.pkg.wordcou nt	WordCountDriver	Driver class to launch the wordcount job
	WordCountMapper	Mapper class for WordCount job. Mapper output: **Key**: State name , **Value**: 1
	WordCountReducer	Reducer class for WordCount job. The reducer will iterate through the values of each state and will calculate total number of times a state has appeared. Reducer output: **Key**: State name , **Value**: Total number of occurrences
	LaunchWordCount	This class is used to launch the WordCount job
com.pkg.mapdata	ReadMapRedOutput	This class is used to read the output from the WordCount job, calculate the percentages and dynamically write those values to a JavaScript file.

com.pkg.sentiment	SentimentAnalysisDriver	Driver class to launch sentiment analysis job
	SentimentAnalysisMapper	Mapper class, which will read each line from the input file and assigns a sentiment rating for each tweet. Mapper output: **Key**: Sentiment rating, **Value**: 1
	SentimentAnalysisReducer	Reducer class will iterate through the values of each sentiment rating and counts the number of rating. Reducer output: **Key**: Sentiment rating, **Value**: Number of tweets with that particular rating
	SentimentClassifier	This is the classifier file obtained from LingPipe APIs, this file will be used by the map reduce job to classify tweets.
	LaunchSentimentAnalysis	This class is used to launch Sentiment Analysis Job
	GeneratePieChart	This class is used to generate a pie chart to display sentiment analysis results
Front end files	main.jsp	Application start page, where user can enter keyword and number of tweets and perform analysis
	satehood.html	When user generates a heat map, they are redirected to this page which will display the tweet density information on a map. colorbrewer.css, example.css, fips.js, polymaps.js, protodata.js, statehood.js, and values.js are the files that support this HTML file.

Functional Overview:

Below is the application start page:



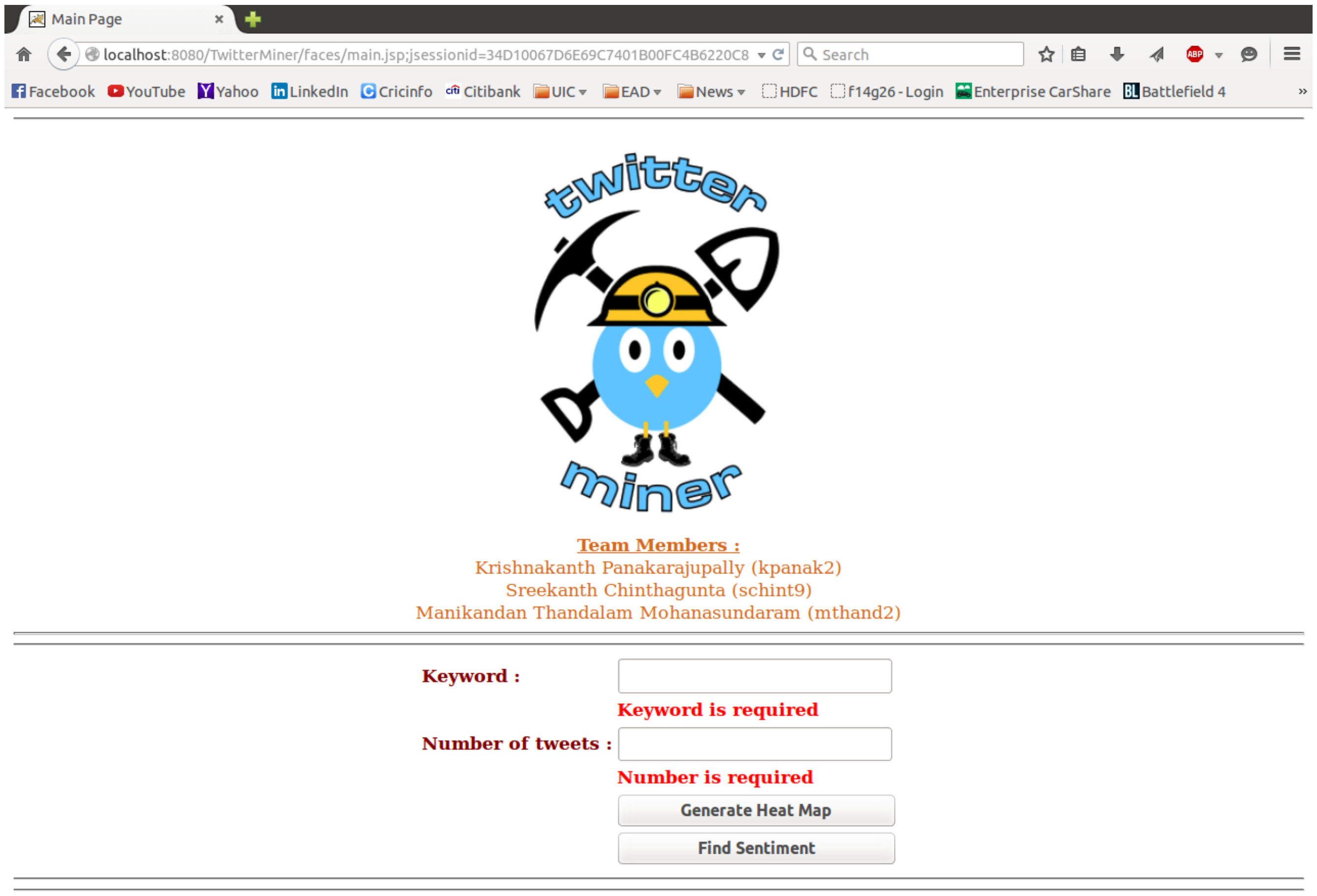
Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword :

Number of tweets :

User can enter any keyword and number of tweets they want to analyze and then either generate a heat map or find the sentiment associated with that keyword.

Both the fields are mandatory and user will not be able to perform analysis without entering some value in the fields:

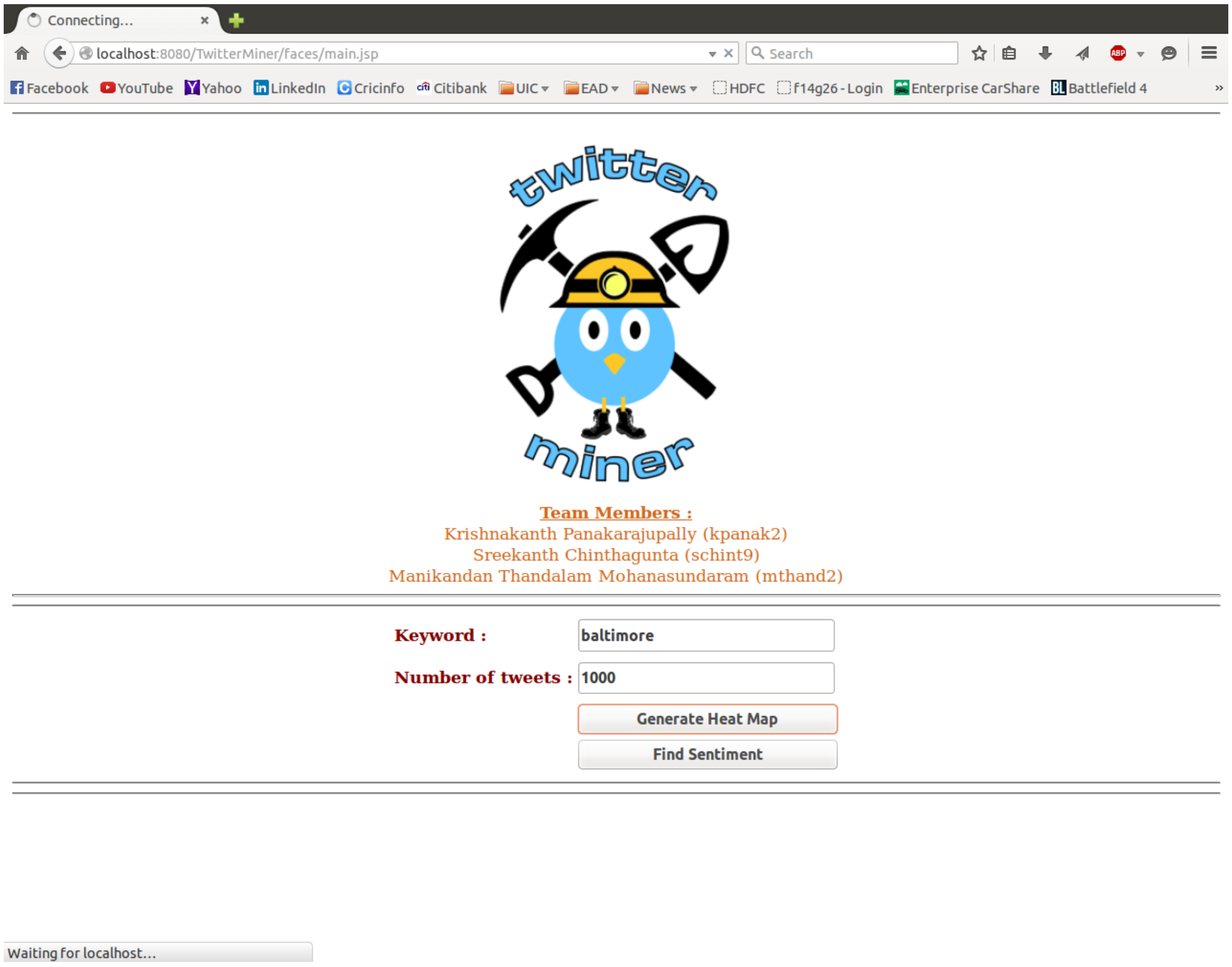


Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword :
Keyword is required
Number of tweets :
Number is required

Generate Heat Map:

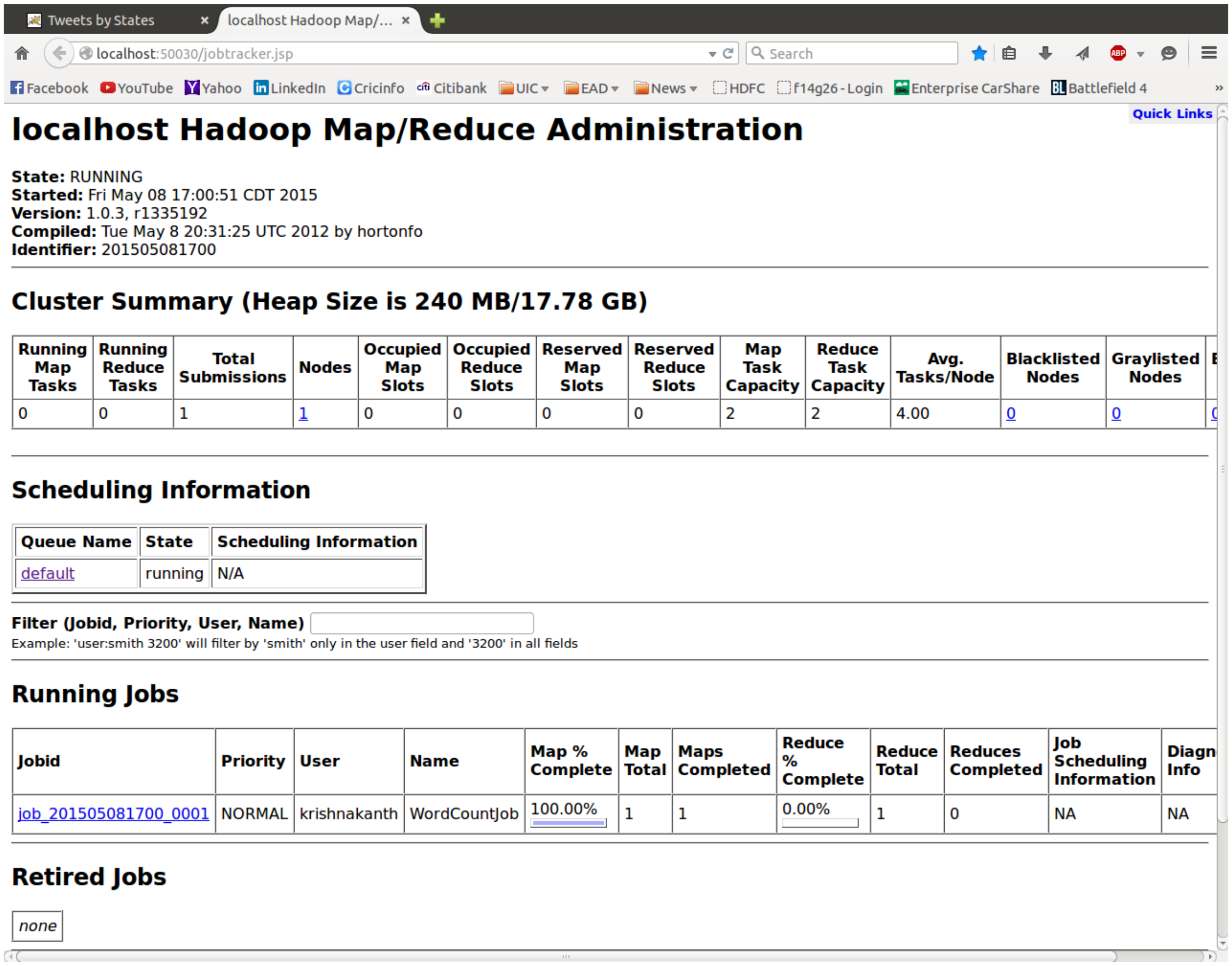
Below is a screen shot showing a user entering keyword "baltimore" and number of tweets as 1000.


Connecting...
localhost:8080/TwitterMiner/faces/main.jsp
Search
Facebook YouTube Yahoo LinkedIn Cricinfo Citibank UIC EAD News HDFC f14g26-Login Enterprise CarShare Battlefield 4
twitter miner
Generate Heat Map
Find Sentiment
Waiting for localhost...

Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword :	baltimore
Number of tweets :	1000

A Map Reduce job is submitted in the background:


Tweets by States
localhost Hadoop Map/...
localhost:50030/jobtracker.jsp
Search
Facebook YouTube Yahoo LinkedIn Cricinfo Citibank UIC EAD News HDFC f14g26-Login Enterprise CarShare Battlefield 4
Quick Links

localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Fri May 08 17:00:51 CDT 2015
Version: 1.0.3, r1335192
Compiled: Tue May 8 20:31:25 UTC 2012 by hortonfo
Identifier: 201505081700

Cluster Summary (Heap Size is 240 MB/17.78 GB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes
0	0	1	1	0	0	0	0	2	2	4.00	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

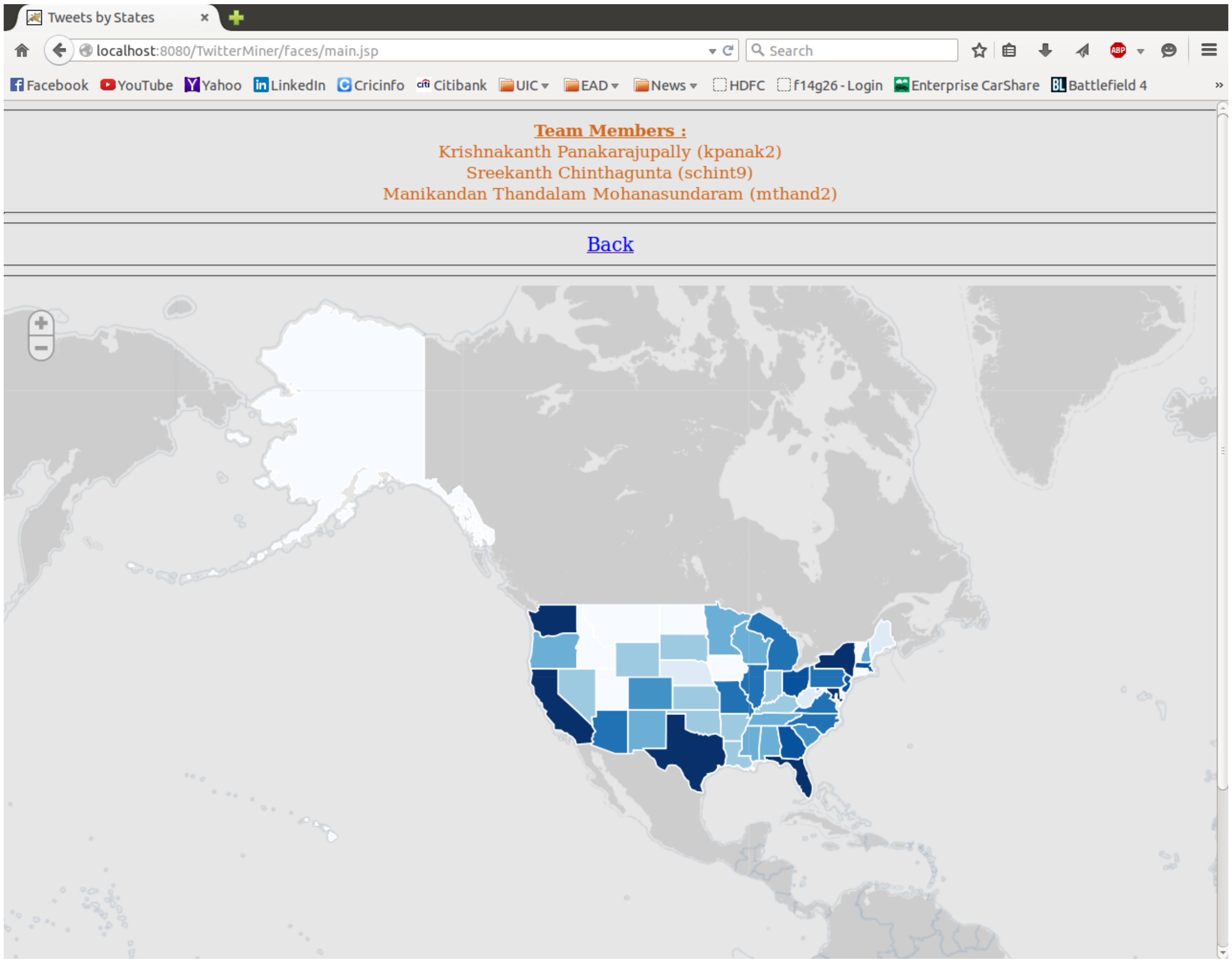
Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagn Info
job_201505081700_0001	NORMAL	krishnakanth	WordCountJob	100.00%	1	1	0.00%	1	0	NA	NA

Retired Jobs

none

Once the Map Reduce job is finished, the user is redirected to statehood.html:



Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

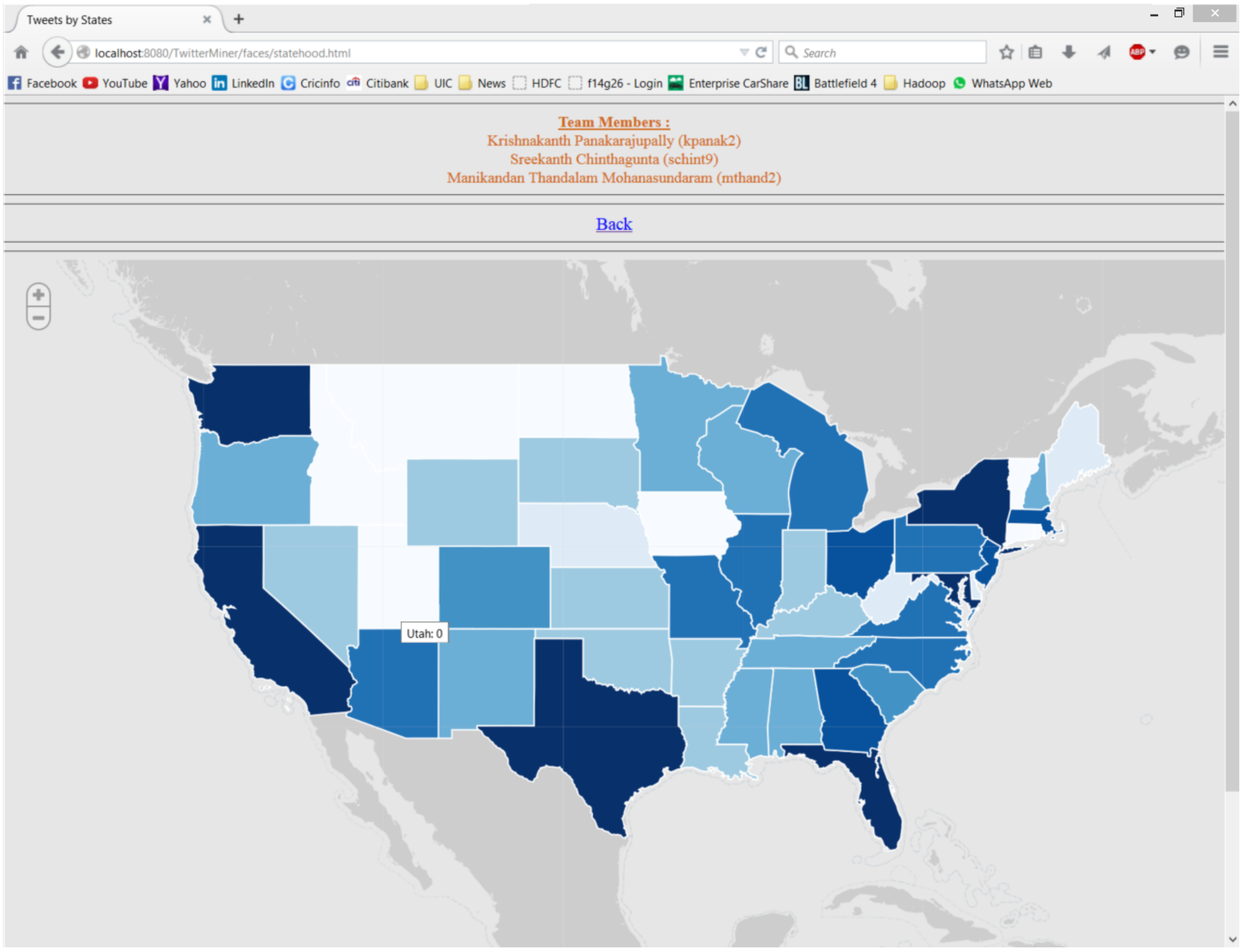
Back

Lighter colors imply low tweet density and darker color imply higher tweet density.

User can place the cursor on any state, to view the percentage of tweets from that state as a tool tip.

Below are some screenshots showing the percentage of tweets from different states:

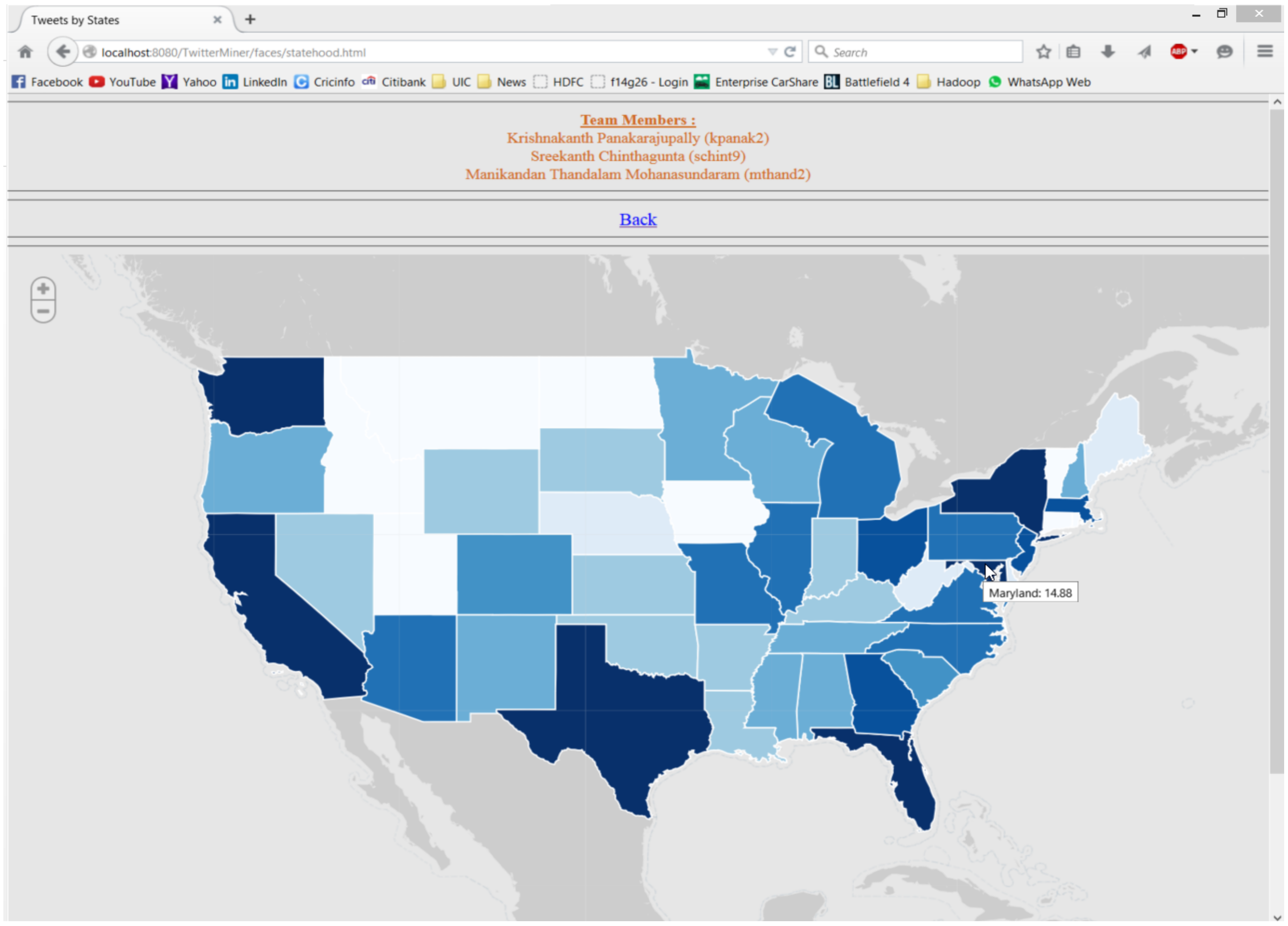
Utah: 0%



Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

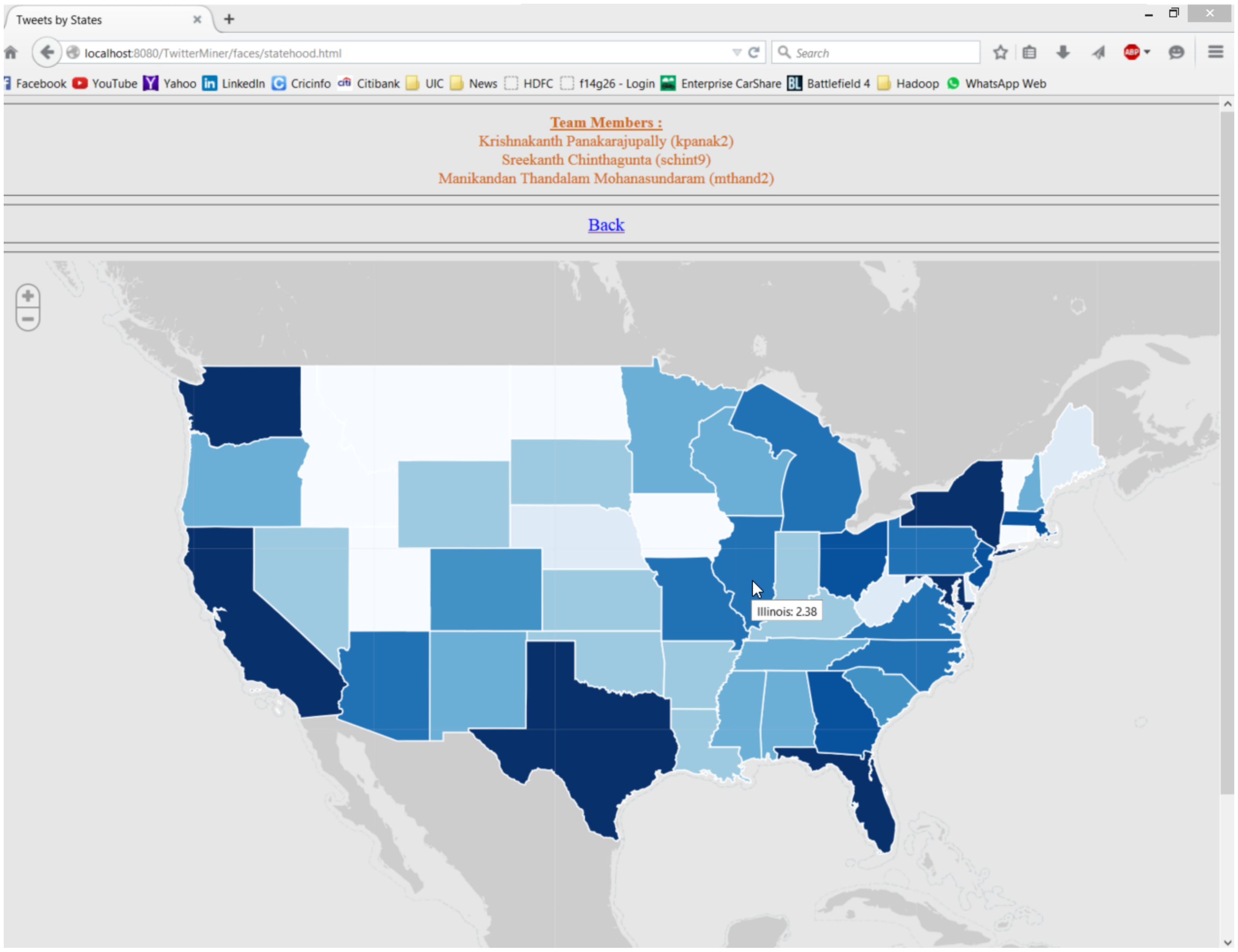
Back

Maryland: 14.88%


Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

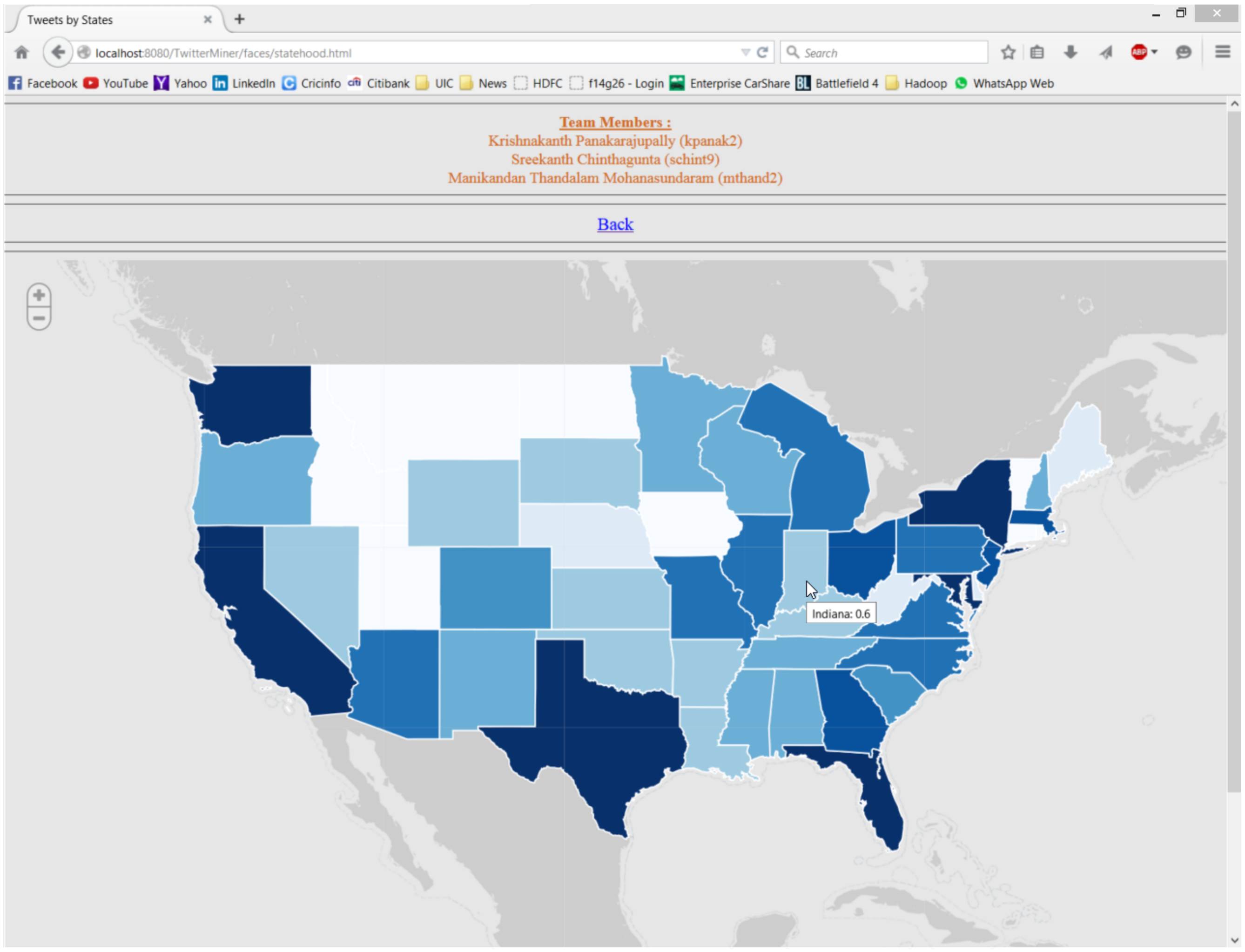
Back

Illinois: 2.38%


Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

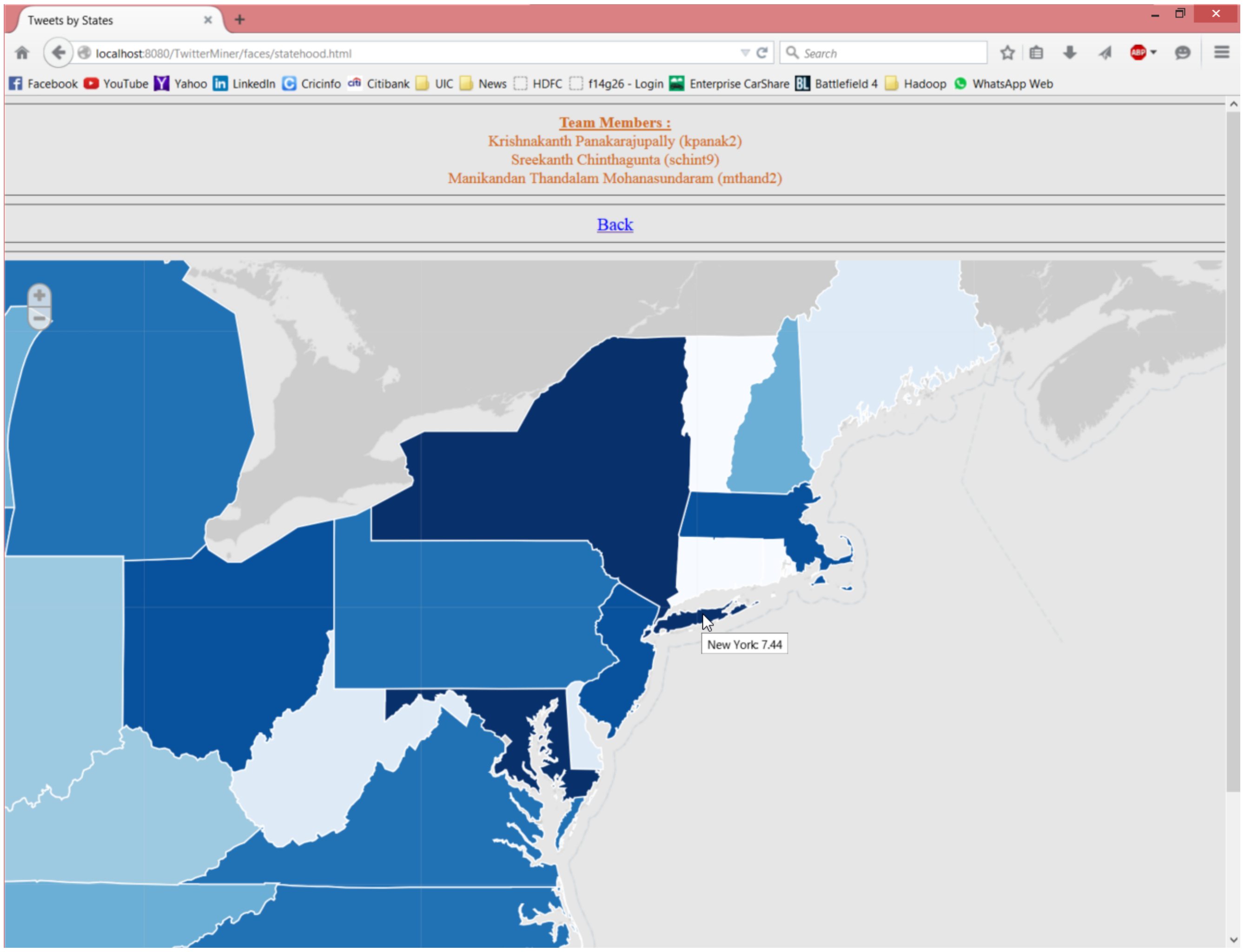
Back

Indiana: 0.6%



Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

User can zoom in, zoom out and pan across the map to view the information.


Tweets by States

localhost:8080/TwitterMiner/faces/statehood.html

Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Back

Below are files generated in the background during the analysis:

File containing the location of each tweet:

data.txt

File from Map Reduce (WordCount) output:

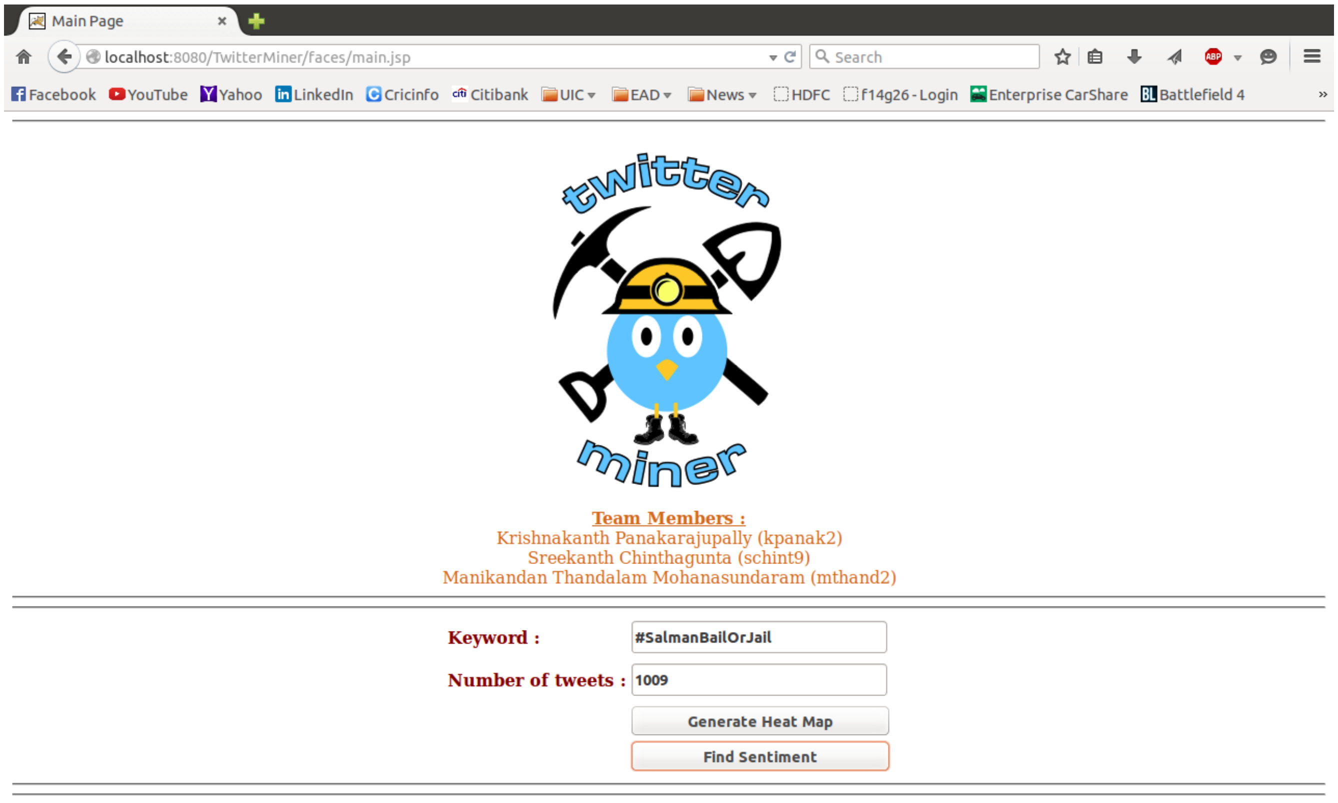
part-r-00000

Java script file containing the percentages:

values.js

Find Sentiment:

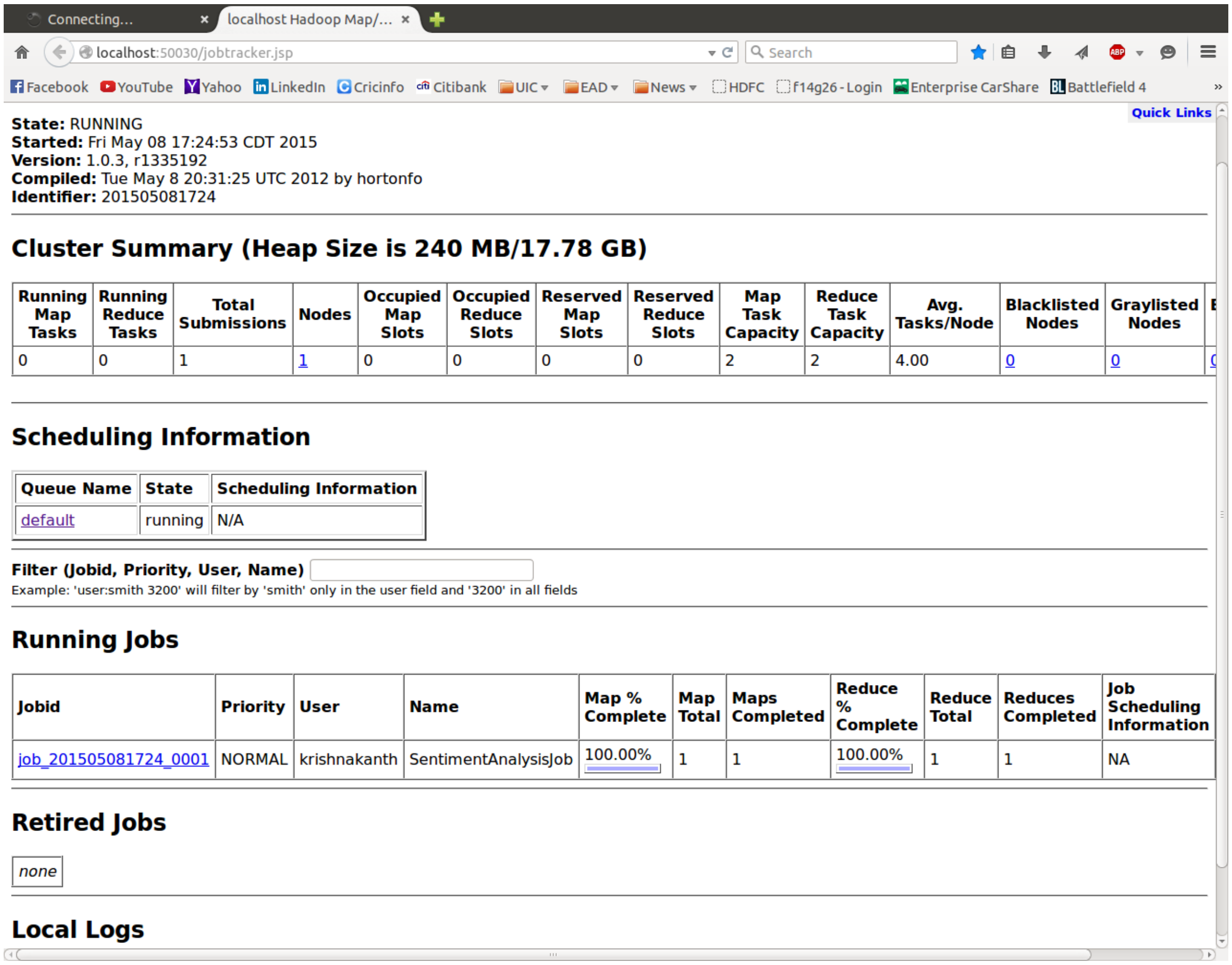
Below is the screen shot showing a user entering “#SalmanBailOrJail” as keyword and 1009 as number of tweets.



Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword : #SalmanBailOrJail
Number of tweets : 1009

A Map Reduce job to perform sentiment analysis is submitted in the background:



State: RUNNING
Started: Fri May 08 17:24:53 CDT 2015
Version: 1.0.3, r1335192
Compiled: Tue May 8 20:31:25 UTC 2012 by hortonfo
Identifier: 201505081724

Cluster Summary (Heap Size is 240 MB/17.78 GB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes
0	0	1	1	0	0	0	0	2	2	4.00	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

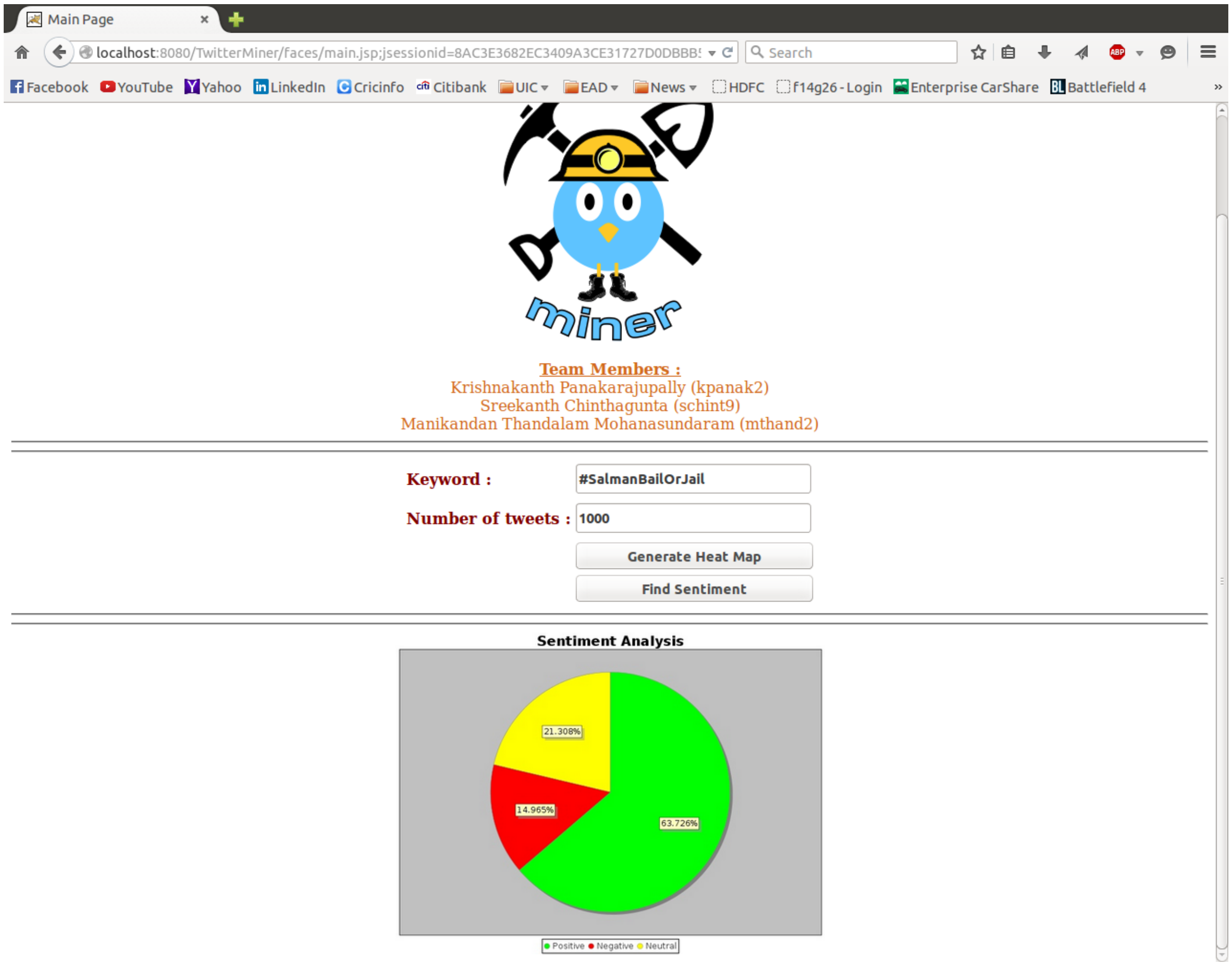
Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201505081724_0001	NORMAL	krishnakanth	SentimentAnalysisJob	100.00%	1	1	100.00%	1	1	NA

Retired Jobs

none

Local Logs

Below is the sentiment analysis result:


Main Page
localhost:8080/TwitterMiner/faces/main.jsp;jsessionid=8AC3E3682EC3409A3CE31727D0DBBB!
Search
Facebook YouTube Yahoo LinkedIn Cricinfo Citibank UIC EAD News HDFC f14g26-Login Enterprise CarShare Battlefield 4
miner

Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword : #SalmanBailOrJail

Number of tweets : 1000

Generate Heat Map
Find Sentiment
Sentiment Analysis
21.308%
14.965%
63.726%
● Positive ● Negative ● Neutral

Files generated during the analysis:

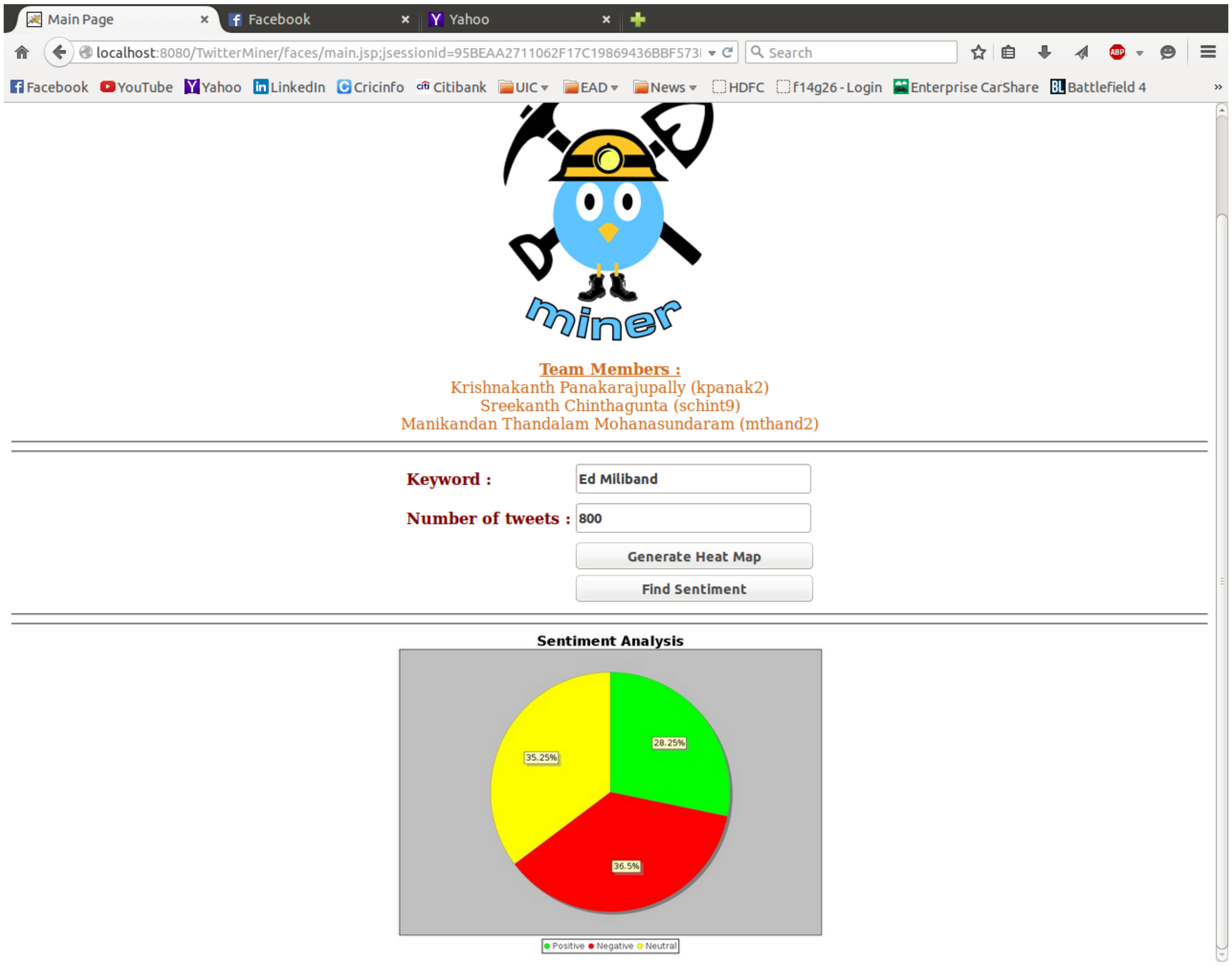
Tweets information (containing 1009 tweets, 1 tweet in each line):

tweets.txt

Map Reduce (Sentiment Analysis) output:

part-r-00000

Below is screen shot of another sentiment analysis performed for keyword “Ed Miliband”


Main Page
Facebook
Yahoo
localhost:8080/TwitterMiner/faces/main.jsp;jsessionid=95BEAA2711062F17C19869436BBF573
Search
Facebook YouTube Yahoo LinkedIn Cricinfo Citibank UIC EAD News HDFC f14g26 - Login Enterprise CarShare Battlefield 4
miner
Generate Heat Map
Find Sentiment
Sentiment Analysis
28.25%
35.25%
36.5%
● Positive ● Negative ● Neutral

Team Members :
Krishnakanth Panakarajupally (kpanak2)
Sreekanth Chinthagunta (schint9)
Manikandan Thandalam Mohanasundaram (mthand2)

Keyword : Ed Miliband

Number of tweets : 800

Conclusion:

As the core functionalities of the application are implemented using Hadoop (MapReduce), the application can be easily scaled to an enterprise level to perform analysis on huge amounts of data, by using cluster computing. The only bottle neck in the current application is the input data from twitter. By getting access to use Twitter Firehose, this application can be made more effective to use at enterprise level.

Work Distribution:

Sreekanth Chinthagunta: Data collection and application design

Krishnakanth Panakarajupally: Front end application development and integration

Manikandan Thandalam Mohanasundaram: Map Reduce algorithms and documentation

References:

Twitter4j: http://twitter4j.org/en/code-examples.html

Polymaps: http://polymaps.org/

LingPipe: http://alias-i.com/lingpipe/