Project Report on

Twitter Data Analysis Using FLUME & HIVE on Hadoop Framework

Developed by,
Jayesh Kumar Sinha under Ardent Computech.

Sixth Semester

B. TECH DEGREE
In
COMPUTER SCIENCE & ENGINEERING
Meghnad Saha Institute of Technology.
2014-2018.

Operation Environment

Operating System: Ubuntu 16.04

Ram: 4 GB or more.

Hard Disk: 100 GB or more

JRE and JDK version 8

Tools and frameworks used:

- Hadoop.
- Apache Flume.
- Hive.

INTRODUCTION

Micro blogging today has become a very popular communication tool among Internet users. Twitter, one of the largest social media site receives millions of tweets every day on variety of important issues. Authors of those messages write about their life, share opinions on variety of topics and discuss current issues. These posts analysis can be used for decision making in different areas like government, Elections, Business, Product review etc. Also sentiment analysis is one of the important area of analysis of twitter posts that can be very helpful in decision making.

Performing Sentiment Analysis on Twitter is trickier than doing it for large reviews. This is because the tweets are very short (only about **140 characters**) and usually contain slangs, emoticons, hash tags and other twitter specific jargon. For the development purpose twitter provides streaming API which allows the developer an access to 1% of tweets tweeted at that time bases on the particular keyword. The object about which we want to perform sentiment analysis is submitted to the twitter API's which does further mining and provides the tweets related to only that object.

Twitter data is generally unstructured i.e. use of abbreviations is very high. Also it allows the use of **emoticons** which are direct indicators of the author's view on the subject. Tweet messages also consist of a **timestamp** and the **user name**. This timestamp is useful for guessing the future trend application of our project. **User location** if available can also help to gauge the trends in different geographical regions.

Apache Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

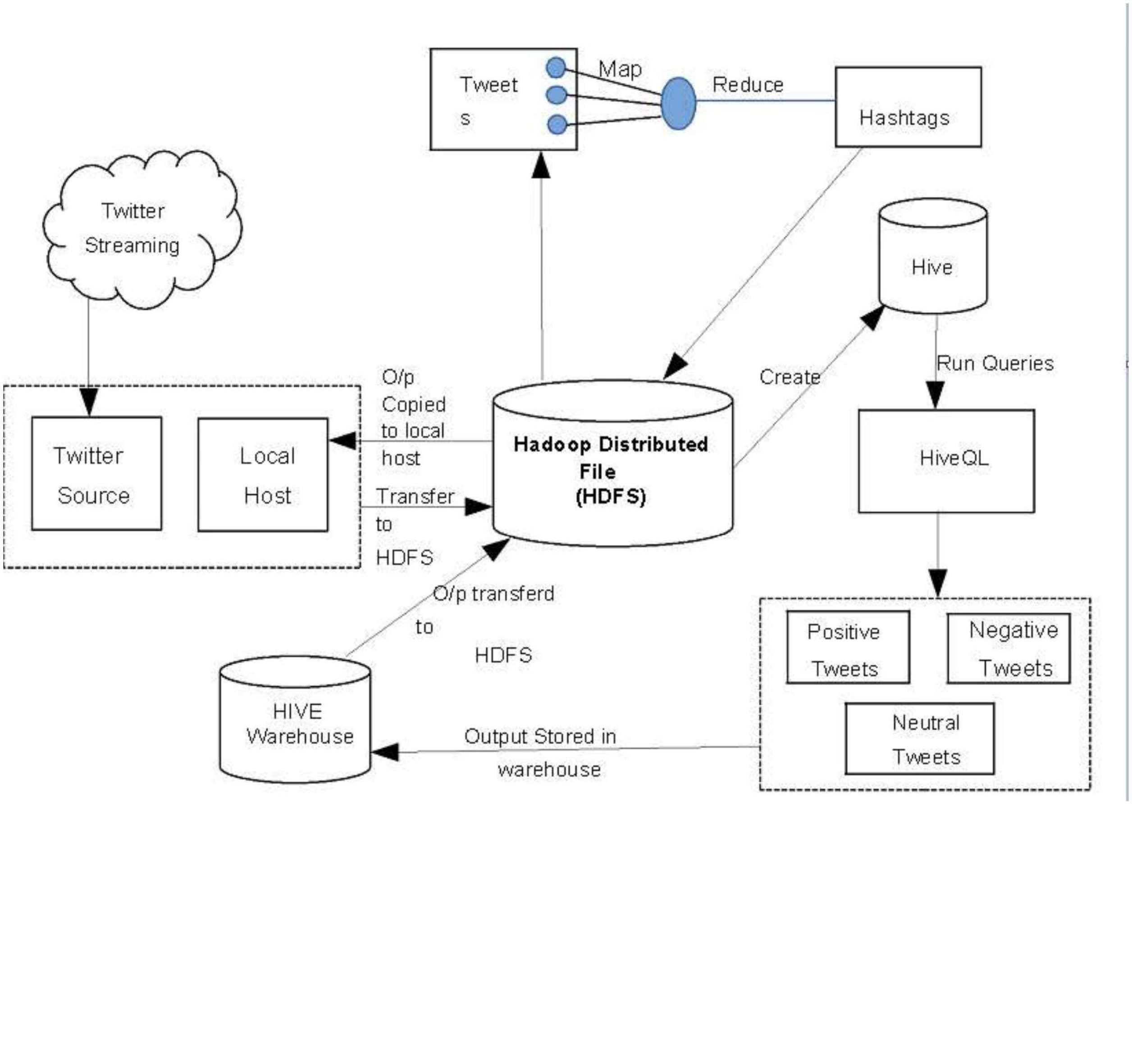
- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

Apache Flume

Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It has a simple and flexible architecture based on streaming data flows; and is robust and fault tolerant with tunable reliability mechanisms for failover and recovery.

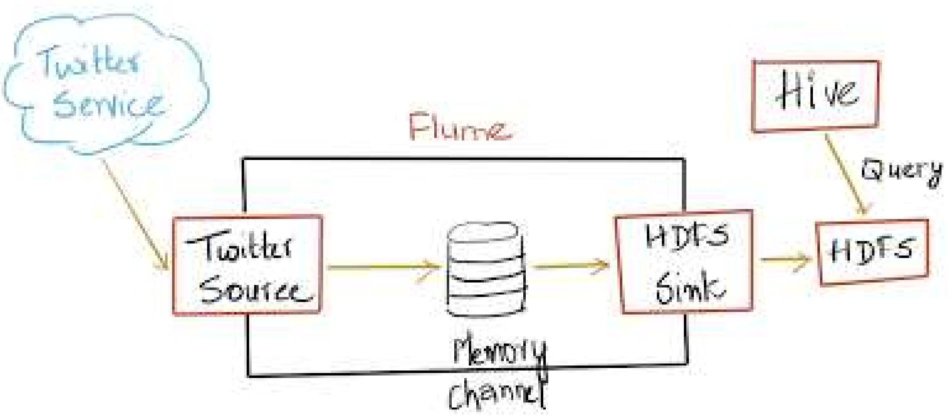
YARN coordinates data ingest from Apache Flume and other services that deliver raw data into an Enterprise Hadoop cluster.

System Architecture



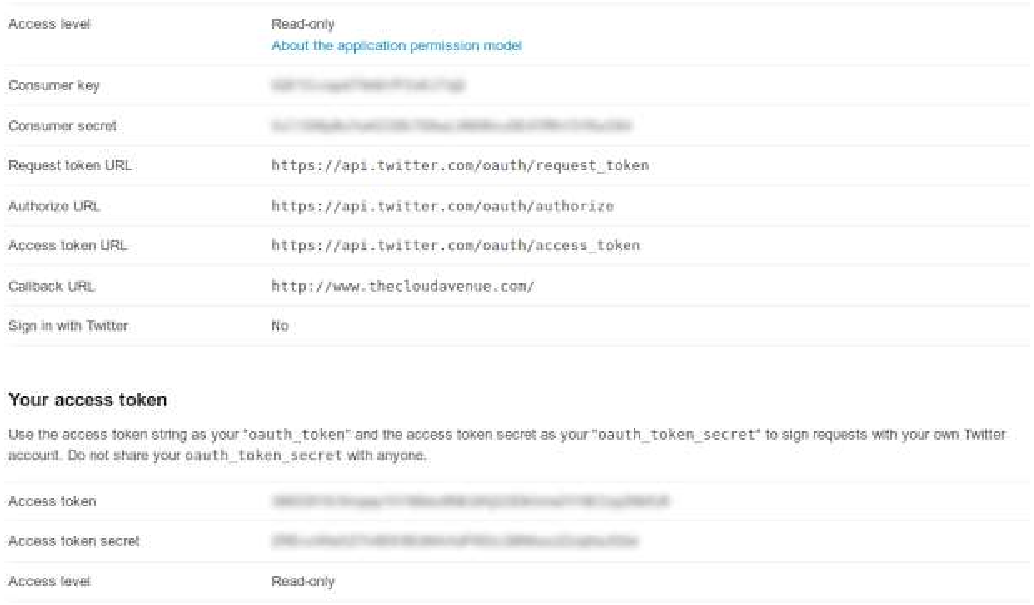
Implementation

Flume has the concepts of agents. The sources, sinks and the intermediate channels are the different types of agents. The sources can push/pull the data and send it to the different channels which in turn will send the data to the different sinks. Flume decouples the source (Twitter) and the sink (HDFS) in this case. Both the source and the sink can operate at different speeds, also it's much easier to add new sources and sinks. Flume comes with a set of sources, channels, sinks and new ones can be implemented by extending the Flume base classes.



Steps

1) The first step is to create an application in https://dev.twitter.com/apps/ and then generate the corresponding keys.



Access level	Read-only About the application permission model
Consumer key	[redacted]
Consumer secret	[redacted]
Request token URL	`https://api.twitter.com/oauth/request_token`
Authorize URL	`https://api.twitter.com/oauth/authorize`
Access token URL	`https://api.twitter.com/oauth/access_token`
Callback URL	`http://www.thecloudavenue.com/`
Sign in with Twitter	No

Your access token

Use the access token string as your "`oauth_token`" and the access token secret as your "`oauth_token_secret`" to sign requests with your own Twitter account. Do not share your `oauth_token_secret` with anyone.

Access token	[redacted]
Access token secret	[redacted]
Access level	Read-only

2) Assuming that Hadoop has already been installed and configured.

3) Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It can be used for dumping twitter data in Hadoop HDFS. After the installation of VMWRE and Hadoop for single node next step come the installation of FLUME. For this you need to log in to twitter. After that go to apps on twitter and create an new application. After you agree with all terms and conditions you will got new application. Then set Consumer Key , Consumer Secret , Owner Key and Owner Secret ID . Now access token need to be created. After the creation of access token and refresh

you will get all the 4 information. Now you Go to flume home and download Apache Flume.

Download the flume-sources-1.0-SNAPSHOT.jar and add it to the flume class path as shown below in the conf/flume-env.sh file
FLUME_CLASSPATH="/home/training/Installations/apache-flume-1.3.1-bin/flume-sources-1.0-SNAPSHOT.jar"

This will automatically be dumped in downloads. Store in desired library. You need to go to apache flume, then go to downloads and extract here and place it In bin and lib. After this you need to configure the file Flume-twitter.conf . The flume.conf should have all the agents defined as below:

```
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 = <consumerKey>
TwitterAgent.sources.Twitter.consumerSecret = <consumerSecret>
TwitterAgent.sources.Twitter.accessToken = <accessToken>
TwitterAgent.sources.Twitter.accessTokenSecret = <accessTokenSecret>
TwitterAgent.sources.Twitter.keywords = Narendra Modi, BJP, Election
TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/user/flume/tweets/
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.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transaction Capacity = 100
```

After a couple of minutes the Tweets should appear in HDFS. If no tweet downloaded in the specified path then refresh. Temporarily data remain in container / Channel and In few seconds tweets start dumping in HDFS. The data downloaded in HDFS is in JSON format. That need to be converted into readable format. Add jsonserde.jar File to convert Json data in readable format.

4) HIVE

Hive is a data warehouse infrastructure tool to process structured data in Hadoop . It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy. Apache Hive (HiveQL) with Hadoop Distributed file System is used for Analysis of data. Hive provides a SQL-like interface to process data stored in HDP. Due its SQL-like interface, Hive is increasingly becoming the technology of choice for using Hadoop. To set up HIVE in Hadoop.
Before we can query the data, we need to ensure that the Hive table can properly interpret the JSON data. By default, Hive expects that input files use a delimited row format, but our Twitter data is in a JSON format, which will not work with the defaults. And we can use the Hive SerDe interface to specify how to interpret what we've loaded. SerDe stands for Serializer and Deserializer, which are interfaces that tell Hive how it should translate the data into something that Hive can process.

To build the hive-serdes JAR, from the root of the git repository:

```
$ cd hive-serdes
$ mvn package
$ cd ..
```

This will generate a file called hive-serdes-1.0-SNAPSHOT.jar in the target directory.
After that Create the Hive directory hierarchy using command below:

```
$ sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse
$ sudo -u hdfs hadoop fs -chown -R hive:hive /user/hive
$ sudo -u hdfs hadoop fs -chmod 750 /user/hive
$ sudo -u hdfs hadoop fs -chmod 770 /user/hive/warehouse
```

You'll also want to add whatever user you plan on executing Hive scripts with to the hive Unix group:

```
 $ sudo usermod -a -G hive <username>
```

After that Configure the Hive metastore.

The Hive metastore should be configured to use MySQL. Follow these instructions to configure the metastore. Make sure to install the MySQL JDBC driver in /var/lib/hive/lib.

Now you need to create tweet table.

Run hive, and execute the following commands:

```
ADD JAR <path-to-hive-serdes-jar>;



CREATE EXTERNAL TABLE tweets (

 id BIGINT,created_at STRING,

source STRING,

favorited BOOLEAN,

 retweeted_status STRUCT<text:STRING,
user:STRUCT<screen_name:STRING,name:STRING>, retweet_count:INT>,

entities STRUCT<urls:ARRAY<STRUCT<expanded_url:STRING>>,

user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,

 hashtags:ARRAY<STRUCT<text:STRING>>>,

text STRING,

 user STRUCT<screen_name:STRING, name:STRING, friends_count:INT,
followers_count:INT, statuses_count:INT, verified:BOOLEAN, utc_offset:INT,
time_zone:STRING>,
 in_reply_to_screen_name STRING ) PARTITIONED BY (datehour INT) ROW FORMAT
SERDE

'com.cloudera.hive.serde.JSONSerDe' LOCATION '/user/flume/tweets';
```

Now you have your data in relational form which can be easily analyzed. Actually this data looks in relational form bur is not. Data is analyzed using Map-Reduce form. Now Queries can be fired on this data for analysis. I collected data for Narender Modi , BJP and election using tweets in tweets table. I want to access 12 most common hashtags on the data.

For this I fired the query :

```
 SELECT LOWER(hashtags.text), COUNT(*) AS total_count FROM tweets LATERAL
VIEW EXPLODE(entities.hashtags) t1 AS hashtags GROUP BY
LOWER(hashtags.text) ORDER BY total_count DESC LIMIT 12;
```

Results in:

```
 Narendra Modi 38890
 BJP 22122
Election 21232
Campaigning 18176
Congress 17656
Votes 18111
Centre 15034
Delhi 11390
```

CONCLUSION

This project will give us hands on experience of handling and parallel processing of huge amount of data. Data collection process will introduce us to Java twitter streaming API. We will get exposure to work with prominent parallel data processing tool: Hadoop.

Apache Hadoop framework is gaining significant momentum from both industry and academia as the volume of data to analyze growth rapidly. This project will help us not only to gain knowledge about installation and configuration of Hadoop distributed file system but also map reduce programming model. Amongst the many fields of analysis, there is one field where humans have dominated the machines more than any – the ability to analyses sentiment, or sentiment analysis.

The future of this data analysis field is vast. This project not only analyses the sentiments of the user but also computes other results like the user with maximum friends/followers, top tweets etc. hence Hadoop can also be effectively used to compute such results in order to determine the current trends with respect to particular topics. This can be very useful in the marketing sector.