NETLIFY TWITTER DATA ANALYSIS

Created By : Lavanya Bhimavarapu

Date : 08/26/2018

Description :

Build an ETL pipeline to collect Twitter Data for :

1. All users that follow @netlify

2. All tweets with #JANstack, #netlify , #CMS

3. Follower-following relationship

SYSTEM DESIGN

1. Data Ingestion :

Twitter data is brought into HDFS using Apache Flume as the data ingestion tool. Flume brings unstructured Twitter data into HDFS Sink using Flume Agent, in this case it will be the TwitterAgent. Twitter data is in the form of JSON tweets. Sample Twitter raw data is provided in Raw\_Data\_Sample.txt file.

The TwitterAgent has Twitter as the Source, MemChannel as the buffer and HDFS as Sink. The setup can be configured using the file twitter.conf in Flume Files folder.

Flume job can be run incrementally and data is staged in HDFS.

Scripts and Setup : Flume Files folder

1. Data Storage :

HDFS is used to store data in the form of JSON files at the location /user/cloudera/netlify\_tweets. Sub-directories are created by the flume agent based on dates to partition the data. By storing input data as partitions, data processing can be manageable and faster.

Creating partitions can be specified in the twitter.conf file.

The data that is brought in is filtered by keywords like ‘#CMS’,’#netlify’,’#JAMstack’,’@netlify’,’Netlify’ to avoid unecessary feeds. Filtering data as early as possible in the ETL process increases throughput later.

Sample HDFS files : Twitter Data folder

1. Data Processing :

HQL is used for creating tables and queries on the staged twitter data. HIVE tables are created using the JSON SERDE. The steps to create and add the jar file is in hivecreatetable.txt file. This file also contains the tables created for storing users information, tweets and user- follower relationships from raw data.

Hive is used due to the support it provides for unstructured data like JSON by using powerful serdes to easily query them.

The Hive tables are created using partitioned external files in HDFS location, so data is brought in directly when new files are added to this location using create table command.

After creating the staging table , data is cleaned and loaded into core tables using insert overwrite :

* netlify\_users
* netlify\_tweets : uses specified tweets
* user\_follower

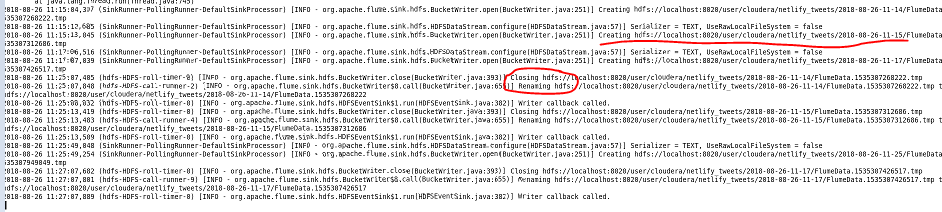
Steps and Setup : hivecreatetable.txt

TECHNOLOGIES USED

1. Cloudera VM 5.13
2. Apache Flume : Data Ingestion into HDFS
3. HDFS : Data Storage for Twitter Data
4. Apache HIVE : Data Processing and Querying using HIVE SQL
5. Oozie : Workflow Management (Not implemented)

LOGGING / MONITORING

Job logging is provided during the data ingestion phase using DFlume logger. The output is displayed in the console. Below is a screenshot of the logs :



The map-reduce tasks can be viewed using HUE on cloudera brower. If any job is taking a long time for execution we can implement some options like repartioning the data or checking the number of reducers or checking for skew data.

Not implemented :

The logs can be filtered and stored in a HIVE table using the run\_id, job\_status and description, which can be queried in case of any errors.

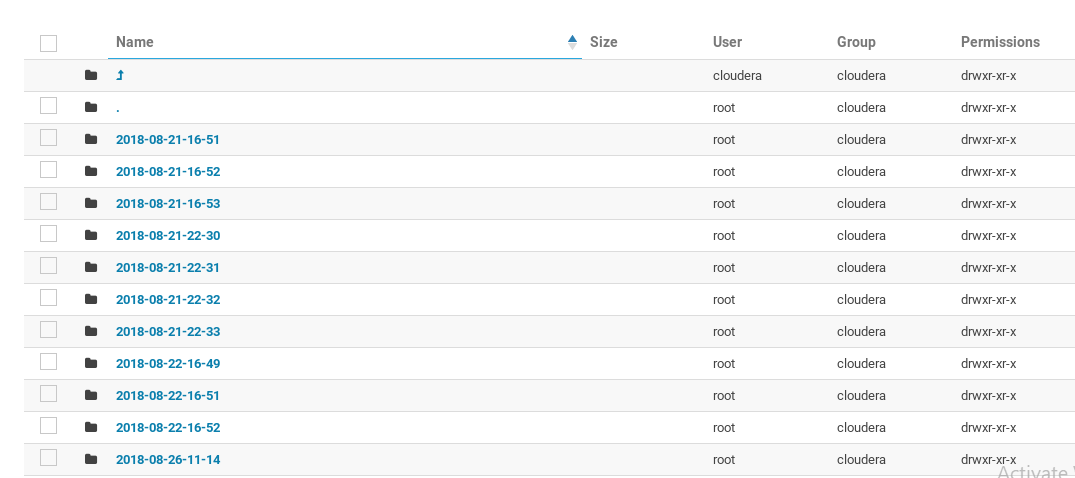
STEPS TO RUN

1. Create an app using Twitter to obtain the consumer and access keys.
2. Download the requires twitter jar files and place it in /etc/flume-ng/lib/
3. Modify twitter.conf using gedit to include below :

* Source : Twitter
* Sink : HDFS
* Channel : MemChannel
* Keywords : #CMS, #netlify, #JAMstack
* Language : en

Twitter.conf file available in Flume Files folder.

1. Run flume agent using flume\_cmd file in Flume Files folder. Can be automated using flume incremental job in Oozie. (Automation not implemented)
2. Check the staged files in HDFS :



1. Create or download a JSON Serde and add the jar to HIVE.
2. Create a hive schema to implement the project.

Create schema if not exists netlify\_project;

1. Setup HIVE configurations and create tables in HIVE using hivecreatetable.txt script.
2. Use test queries in Analysis\_SQL.txt to query HIVE tables.

HIVE SCHEMA

Schema : netlify\_project

Tables :

1. Netlify\_tweets

Staging table that uses HDFS external files to bring data into HIVE. New data is appended using flume and brought over to Hive incrementally.

Table is partitioned by date and uses JSON format.

1. Netlify\_users

Stores information of all Netlify users.

Selects distinct rows of user JSON object from netlify\_tweets for each tweet and inserts into netlify\_users. Information related to user\_screen\_name (Assumed key), follower counts, following counts, so on is stored.

1. Tweets\_info

Selects information related to tweets from netlify\_tweets like time created, tweet id, is it a re-tweet, user who tweeted it and inserts into tweets\_info.

1. User\_follower

This table can be used to maintain follower-following relationship. New users are added along with the followers of that user into the table. For each (user, follower) key we have additional columns like timestamp of started following, timestamp of unfollow to support queries.

Flag column can be added to denote if the user-follower relationship is active or inactive to improve query performance. (Not implemented)

Note : Staging table netlify\_tweets doesnot contain all columns from twitter JSON. Only selected columns related to queries are brought over. Additional columns can be added by modifying create table script.