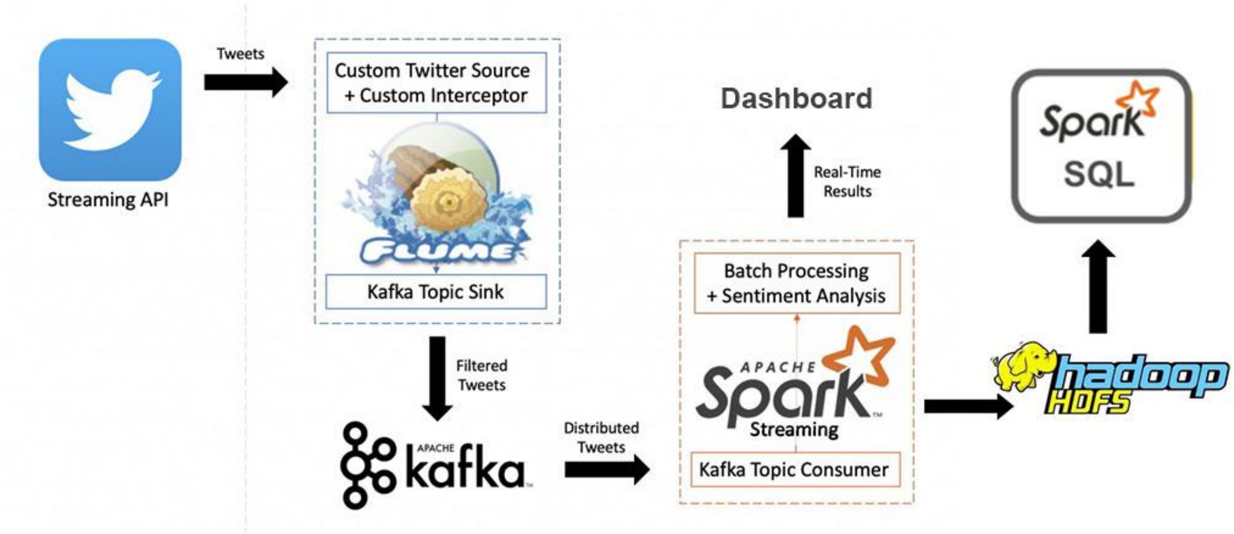
REAL-TIME TWITTER DATA ANALYSIS

Course project
Big Data Technology (CS523)
Maharishi International University
March 2020

Team members
Van Nhat Khong 611039
Wina Tesfamariam Merhazion 610591
Muhammad Luqman 611023

Description

This application is created using the Twitter API to stream data and create real time analysis of tweets from Twitter. The application is able to gather and filter real-time data and displaying the results in a matter of seconds. Meanwhile, it stores data to HDFS system in a clean format for historical data analysis. Not only showing most mentioned topics and users, the application also provides information about sentiment of tweets by doing natural language analysis.



Technologies used

- Cloudera VM (CENTOS 6)
- Twitter API 7.0
- Flume 1.6.0
- Kafka 2.12

- Spark 2.4.5 (SparkSQL, Spark Streaming)
- Hive 8.0
- Flask 1.1.2
- Python 2.7
- Textblob 0.8.4

0. Project Setup

This phase included deciding on a design that would fulfil our idea. this included deciding the technologies that were used. Since our application would need to receive data from Twitter we need to acquire a token, that would enable our application to access the twitter API.

1. Flume and Kafka

We used Flume to get the incoming data and from a twitter source and have a sink as a Kafka-Sink

-- Starting flume

```
flume-ng agent --conf /etc/flume-ng/conf \
--conf-file /home/cloudera/Desktop/FinalProject/Flume/flume_twitter_to_kafka.conf \
--name KafkaAgent \
--plugins-path /home/cloudera/Desktop/FinalProject/Flume/ \
-Dflume.root.logger=INFO,console
```

-- Starting Zookeeper Server

```
cd /opt/kafka_2.12-2.4.1/
bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
```

-- Start Kafka server

```
cd /opt/kafka_2.12-2.4.1/
bin/kafka-server-start.sh config/server.properties
```

--Start Kafka producer

```
cd /opt/kafka_2.12-2.4.1/
bin/kafka-topics.sh --zookeeper quickstart.cloudera:2181 --create --topic twitter_stream --partitions
1 --replication-factor 1
```

--Start Kafka consumer

```
cd /opt/kafka_2.12-2.4.1/
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic twitter_stream --from-beginning
```

2. Spark Streaming

After receiving data from flume we were able to create a custom parser. Furthermore, we were able to infiltrate the English words and output incoming data to both HDFS and to the dashboard.

-- Start Spark streaming

```
cd /home/cloudera/Desktop/FinalProject/SparkStreaming/TwitterStreamAnalysis
export SPARK_HOME=/usr/local/spark/
export SPARK_LOCAL_IP=127.0.0.1
spark-submit --master 'local[3]' --jars /home/cloudera/Desktop/FinalProject/SparkStreaming/jars/
trending_topics_sentiment.py twitter_stream
```

3. SparkSQL and Hive

We were able to retrieve data from SparkSQL using Hive. We were able to query the user count and average user sentiments. many more queries could be implemented though this abstraction.

4. Visualization

We were able to perform visualization to the dashboard using Flask and Python. here we could see the positive, negative and neutral sentiments outputted in real time. Furthermore, we were also able to display the top ten hashtags and most mentioned users to the dashboard.

-- Start Dashboards

```
cd /home/cloudera/Desktop/FinalProject/Dashboard/TwitterAnalysisDashboard
flask run -p 5001
```

5. Further Development

- Publish our web application with additional charts that provide more information relating to society’s trends and interests with data from Twitter API and other real-time streaming data sources.