ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

REAL-TIME DATA PROCESSING WITH STORM: USING TWITTER STREAMING

Gireesh Babu C N[1], Manjunath T N[2], Suhas V[3]
[1,2,3]Department of Information Science and Engineering, BMSIT&M, Bengaluru, India

DOI: 10.5281/zenodo.822928

ABSTRACT

In today's highly developed world, every minute, people around the globe express themselves via various platforms on the Web. And in each minute, a huge amount of unstructured data is generated. Such data is termed as big data. User opinions are related to a wide range of topics like politics, latest gadgets and products. Here we propose to analyze the sentiments of Twitter users through their tweets in order to extract what they think. We classify their sentiments into three different polarities – "positive", "negative" and "neutral." Since, 6000 tweets are generated every second and this number is increasing, we need a robust system to process these tweets in real-time. Here, batch-processing would have its limitations and therefore a real-time and fault tolerant system, Apache Storm is used. This process as a whole is also called as Opinion Mining or voice of the customer.

KEYWORDS: Object detection, parallel computing, distributed computing, Storm.

I. INTRODUCTION

Earlier, natural language processing was applied to news sites and blogs where the quantity of data was large. But in today's time, micro blogging sites are becoming more popular where length of the post is shorter but the number of posts per day has shot up. Users have lesser number of words to express their opinion and hence tend to convey their sentiments through emoticons and acronyms, making sentiment analysis a more difficult task. We overcome this problem by making use of dedicated dictionaries for lexical words and emoticons associated with their respective polarities. Alongside, two dictionaries are used to identify stop words and acronyms. For instance, when a movie is released, the viewer tweets about his/her opinions.
With the help of our system the director or the marketing team of the movie can modify their marketing strategy according to the generated analysis. In another case, when the customer tweets about a particular product, whether he finds the product appealing or he desires any specific changes, the owner/developer of the product can use our system to get a better understanding of the mass opinion and make any changes if necessary so as to increase the sales. On a more individual front, if a Twitter user wants to analyze his own behaviors or mood over a certain period of time based on his tweets, he can use the system to view his average happiness over that period of time.

Since users around the world regularly tweet about recent happenings (whether in their personal lives or in a public domain), it demands a fast, scalable, reliable and fault-tolerant system that processes tweets continuously. This is the reason we chose our platform to be Apache Storm. We provide input to the Storm cluster which streams the tweets in real time and provides the robust system we need, in order to process the thousands of tweets being generated every second. The fully developed application will provide live statistics. These statistics can be in the form of graphs, charts or relationship maps and will provide easy readability and understandability of user sentiment (whether positive, negative or neutral).Recent advances in sensor technologies, mobile positioning, wireless communication, and social media lead to the production and collection of Big Data[5] in the form of continuous and dynamic data streams.

Such data with high arrival rate must be continuously and efficiently collected, integrated, processed and analyzed, in order to provide support for timely decisions in many areas such as transportation, emergency management, healthcare, environment monitoring, and energy. This opens up new challenges and opportunities to address the key aspects of mobile sensor-based big streaming data, namely, volume, velocity, variety, and

veracity. Deployment and maintenance of a sensor network for a particular purpose is a hard, expensive and time-consuming task. Recently, the focus in wireless sensor network research has shifted from static networks of specialized sensor devices deployed in the environment, to a people-centric approach relying on the mobility of people and their mobile devices. The idea of allowing mobile users to collect, process and share the information gathered by their mobile devices leads to mobile crowd sourcing/sensing paradigm.

Owing to the exponential growth and availability of smart phones and tablet devices, massive structured and unstructured real-time streaming data is generated, collected by these devices and sent to the servers/cloud. Such data is geographically and temporally referenced, such as mobile GPS location streams, geo-tagging Twitter messages, spatio-temporal photo/image streams, etc. To efficiently collect, process, mine, and analyze that massive amount of real-time spatio-temporal data streams in a timely manner the use of high performance computing methods and techniques over computer cluster/cloud infrastructure is needed.

II. RELATED WORK

Big Data processing in real time [5] can be done using many tools such as apache flink [4], apache spark[6] and apache storm. Here apache storm is used to perform real time analysis. Related to our problem, the existing work explained by taking few examples as under:

[1] Twitter Sentiment Analysis using Apache Storm IEEE 2014
However, exponential growth of real-time applications in a large scale, where computations can be divided into independent tasks, urges a new computing paradigm called real-time stream processing platform. This type of platform is doing for real-time processing what Hadoop did for batch processing, facilitating the unbounded real-time stream processing of data. Scalability, fault-tolerance and data guarantee are the desired characteristics of this computing platform.

[2] Processing Big Trajectory and Twitter Data Streams using Apache STORM IEEE 2015
Storm--developed under the Apache License-- is the pioneers of real-time stream processing systems, which is vastly applied in big data solutions Storm is a distributed real-time computation system build for intensive data processing, when the 9 computation model fit streaming process, it provide the best performance.

[3]Apache Storm Based on Topology for Real-Time Processing of Streaming Data from Social Networks IEEE 2016
Processing high volumes of data collected over a short period of time requires efficient, scalable data storage and computing platform. A batch processing system contains Map/Reduce and is generally more concerned with size and complexity of jobs than latency of computation.

III. METHODOLOGY

3.1 PROPOSED ALGORITHM

Real-time can be measured in milliseconds. While having second or millisecond latency is not crucial to the end user — it does have a significant effect on the overall processing time and the level of analysis that we can produce and push through the system. As many of those analyses involve thousands of operations to get to the actual result.
Data arriving in bulk, analyzing them on the go and sorting them to their respective areas is a challenge that is still in the process of being tackled.
Solving problems like this using the abilities we possess and stepping closer to the solution is being prioritized rather than storing and managing them later on.

3.2 DESIGN

Real-time example of "Twitter Analysis" and see how it can be modelled in Apache Storm. Figure 3.1 diagram depicts the structure.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

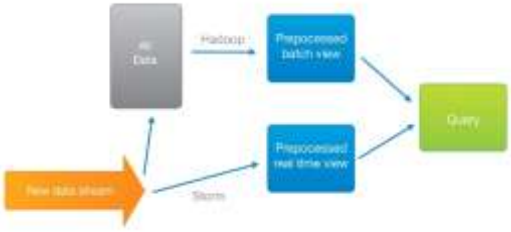


Figure 3.1 How data will be Analysis in Real-Time

In Figure 3.1 Show how real data can be performed and shows the difference between the Hadoop and Apache Storm Process

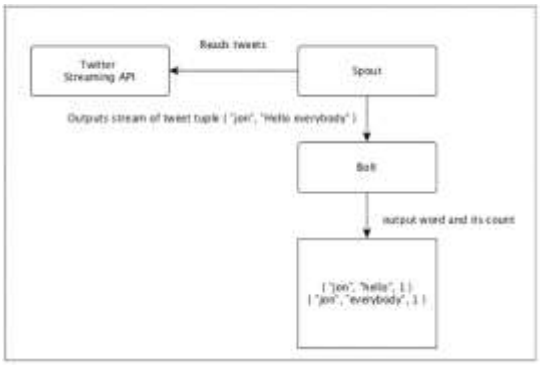


Figure 3.2 Twitter Real-Time Analytics Architecture

Figure 3.2 shows how real time data will be executed in Twitter

Storm – Cluster Architecture

Figure 3.3 Apache Storm-Cluster Architecture

The Figure 3.3 Show the Architecture of the Apache Storm.

Two kinds of nodes

• Master node runs a daemon called Nimbus

• Each worker process executes a subset of a topology

Nimbus Node

This is the master node. It uploads the computation to be performed in the cluster, launches worker nodes and even re-assigns worker nodes in case of failure.

Zookeeper Nodes

These nodes are assigned on every slave machine. The basic function of the zookeeper nodes is to keep a check on the processing happening on the worker nodes..

Supervisor Nodes

There nodes, assigned on every slave machine, start and stop workers according to commands from the nimbus.. A key abstraction in Storm is the topology which is in fact, the program that keeps running in the Storm cluster.

3.3 DATA COLLECTION

Twitter offers three ways to access its data and these are:

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

1. Twitter Search API

Twitter Search API helps to access a data set that exists from tweets previously written. In the Search API, users ask for tweets that match a search criteria or even a part of it. The criteria could be keywords, usernames or locations. But there is a limit to the number of tweets that can be accessed. For an individual user, the maximum number of tweets that can be received is the last 3,200 tweets, regardless of the query criteria. With a specific keyword, Twitter only polls the last 5,000 tweets per keyword.

2. Twitter Streaming API

Twitter Streaming API is a push of data as tweets happen in near real-time unlike the Twitter Search API. This API is free of cost although the percentage of total tweets users receive with the Twitter Streaming API varies heavily based on the criteria users request and also on the traffic. Therefore, use of this API cannot be relied upon to generate a large number of tweets for real-time processing.

3. Twitter Firehose

Twitter Firehose guarantees delivery of 100% of the tweets that match the criteria But the difference lies in the cost. Twitter Firehose is not free of cost, unlike the Streaming API.

Implementation

The spout does not perform any processing on the data. It simply streams it. These tweets are then sent by the spout to the bolt in the cluster so that they can be processed.

Sentiment Classifier

The tweets are broken down into tokens where each token is assigned a polarity, which is a floating point number, in the range from -1 to 1: negative sentiments being -1, neutral being 0 and positive being 1. The overall sentiment is then calculated by adding the polarities of each token. The evaluated sum is then rounded off to the nearest integer, thus assigning the final polarity to the tweet.

How Tweets will be selected

Positive Tweets:

Tweets indicating an achievement or celebration or congratulating someone Tweets using emoticons such as :) , :D , =) , =D, ^_^ ,

E.g.: Chasing 275 to win, India comfortably reached the target with 28 balls to spare due to some excellent batting from the top order. #IndvsSL

Negative Tweets:

Tweets indicating boredom (E.g. if a movie wasn't entertaining) Tweets indicating dejection (E.g. due to a tragedy in one's personal life) Tweets indicating anger (E.g. due to ongoing riots in an area) Tweets using emoticons such as :(, :'(, :/ , -_-

E.g.: I just lost $203 and I'm so angry! How do you deposit $ into the wrong account after I gave you the correct information! #Angry

Neutral Tweets:

Tweets including both positive and negative reviews Tweets including neither positive nor negative sentiments Tweets presenting facts or theories.

E.g.: I have been studying whole day long. Hope it helps!!!!

After starting Storm user interface application, type the URL **http://localhost:8080** in your favorite browser and you could see Storm cluster information and its running topology.

Running a topology:

storm jar all-my-code.jar backtype.storm.MyTopology arg1 arg2

Killing a topology

storm kill {topology name}

APPLICATION

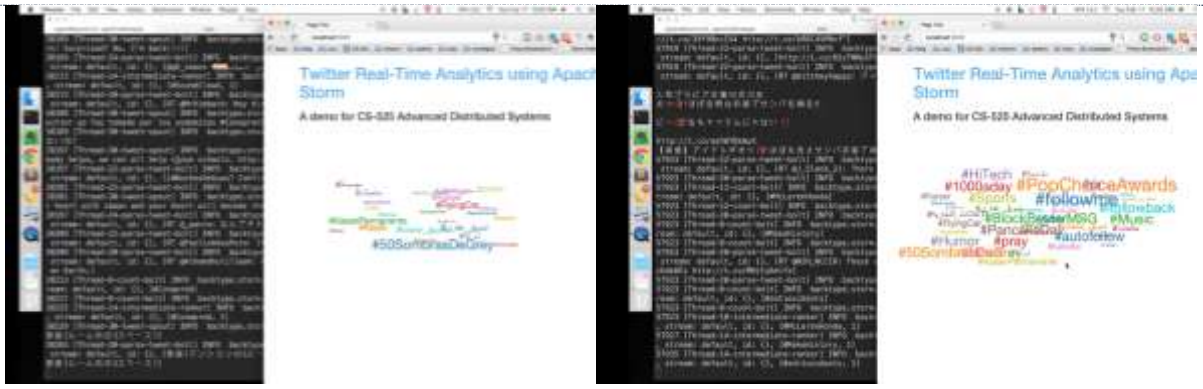
Storm provides a twitter spout, *TwitterSampleSpout***,** in its starter kit. We will be using it to retrieve the tweets. The spout needs OAuth authentication details and at least a keyword. The spout will emit real-time tweets based on keywords.

The tweet emitted by spout will be forwarded to *HashtagReaderBolt*, which will process the tweet and emit all the available hashtags. HashtagReaderBolt uses *getHashTagEntities* method provided by twitter4j. getHashTagEntities reads the tweet and returns the list of hashtag.

IV. EXPERIMENTAL RESULTS

In Figure 4.1(a) once the Code as been executed then the Output will be Displayed in Local Host which you specify in the Storm Configuration.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7



(a) (b)

***Figure 4.1(a) Output will be displayed in the Localhost, (b)* Output will be growing as it refreshed**

The Figure 4.1(b) in that the tweet will be growing large as refreshed and the large tweet indicate that it as been tweeted more number of time.

V. CONCLUSION

A Twitter storm is a sudden spike in activity surrounding a certain topic on the Twitter social media site. A Twitterstorm is often started by a single person who sends his or her followers a message often related to breaking news or a controversial debate. Fast – benchmarked as processing one million 100 byte messages per second per node. Scalable – with parallel calculations that run across a cluster of machines Fault-tolerant – when workers die, Storm will automatically restart them. If a node dies, the worker will be restarted on another node. Reliable – Storm guarantees that each unit of data (tuple) will be processed at least once or exactly once. Messages are only replayed when there are failures. Easy to operate – standard configurations are suitable for production on day one. Once deployed, Storm is easy to operate.

VI. REFERENCES

[1] ”Twitter Sentiment Analysis using Apache Storm”Ishana Raina, SourabhGujar, Parth Shah, Aishwarya Desai, Balaji Bodkhe.
[2] ”Processing Big Trajectory and Twitter Data Streams“using Apache STORM Dragan Stojanović1, Natalija Stojanović2, Jovan Turanjanin3.
[3] “Apache Storm Based on Topology for Real-Time Processing of Streaming Data from Social Networks “Anatoliy Batyuk, Volodymyr Voityshyn Lviv Polytechnic National University, 12 Bandera street, Lviv, 79013, Ukraine.
[4] ”Real Time Big Data Analysis Using Apache Flink”, Anu Pokhrel et.al., International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-3, Issue-6, June 2017
[5] *”*Complex Event Processing In Smart Homes” by Pooja K.S *et al.*,in International Journal of ScientificEngineering and Applied Science (IJSEAS) - Volume-1, Issue-3, June 2015.
[6] Kenny Ballou (2016, Apr 21). Apache Storm vs. Apache Spark [Online] – Available: http://zdatainc.com/2014/09/apache-stormapache- spark.

CITE AN ARTICLE

CN, G. B., N, M. T., & V, S. (2017). REAL-TIME DATA PROCESSING WITH STORM: USING TWITTER STREAMING . *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 6*(7), 6-10. doi:10.5281/zenodo.822928