IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 10, October 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

Streaming Data Analysis using Apache Cassandra and Zeppelin

A. MadhaviLatha[1], G.Vijaya Kumar[2]

[1] PG Student, Computer Science and Engineering Dept, GPREC,
Kurnool (District), Andhra Pradesh-518007, INDIA. madhavilatha.aavula@gmail.com

[2] Assistant Professor, Computer Science and Engineering Dept, GPREC,
Kurnool (District), Andhra Pradesh-518007, INDIA. gvjkumar@gmail.com

Abstract

Big data is a popular term used to describe the large volume of data which includes structured, semi-structured and unstructured data. Now-a-days, unstructured data is growing in an explosive speed with the development of Internet and social networks like Twitter,Facebook & Yahoo etc., In order to process such colossal of data a software is required that does this efficiently and this is where Hadoop steps in. Hadoop has become one of the most used frameworks when dealing with big data. It is used to analyze and process big data. In this paper, Apache Flume is configured and integrated with spark streaming for streaming the data from twitter application. The streamed data is stored into Apache Cassandra. After retrieving the data, the data is going to be analyzed by using the concept of Apache Zeppelin. The result will be displayed on Dashboard and the dashboard result is also going to be analyzed and validating using JSON.

Keywords: *BigData, Unstructured, Hadoop, Flume, Spark Streaming,Twitter, Apache Cassandra, Zeppelin, Analysis, JSON.*

1. INTRODUCTION

In terms of relational databases, moving and modifying large volumes of data into the necessary form for Extraction, Transformation, Loading (ETL) can be both costly and time consuming. To process or analyze this huge amount of data or extracting meaningful information is a challenging task now a days. Big data exceeds the processing capability of traditional database to capture, manage, and process the voluminous amount of data. Falling storage costs and access to better compute power for less money have made the software more attractive as datasets continue to grow, As a result, many companies are rethinking their move toward to traditional enterprise storage and architecture to leverage big data. Hadoop is best suited for processing all types of data. Fig 1 represents all types of data.

Big Data consists of very large volumes of heterogeneous data that is being generated, often, at high speeds. These data sets cannot be managed and processed using traditional data management tools. The size of the data will ranges from terabytes to many petabytes. Big data[1] is an umbrella term that not only refers to the enormous amount of data available today but also the complete process of gathering, storing, retrieving, and analyzing the available data.

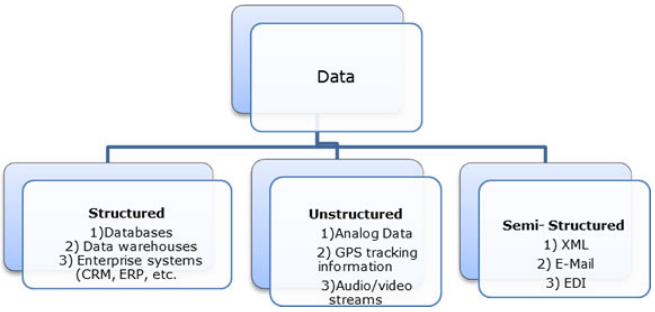

Fig 1: Types of Data

Hadoop[2] was created by Doug Cutting and Mike.Hadoop is a framework from the Apache software foundation written in Java. The motivation comes from Google's Map

Reduce and Google File System. Fig2 represents Apache Hadoop Ecosystem, it describes all tools available in Hadoop.

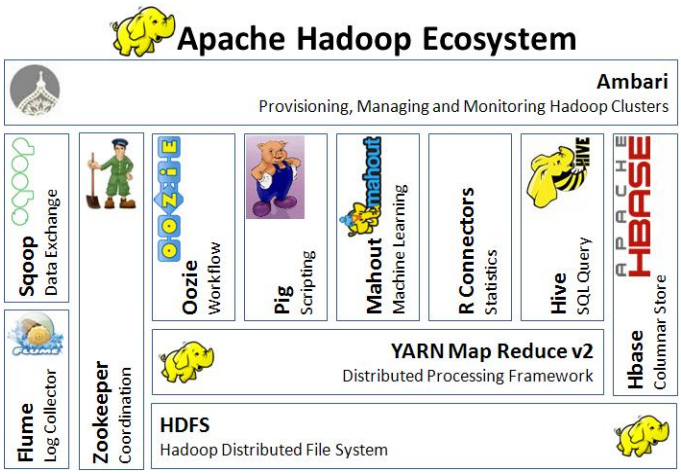

Fig 2: Describes Hadoop Ecosystem

The data can be generated from different sources like social media (Face book, Twitter, and YouTube); sensor data, stock market exchanges, transactional applications etc. A huge amount of data is stored by different kinds of applications for future analysis. Relational databases[5] are used for decades as data storages, although in some cases they are not proper for Big Data processing. To solve this problem, non-relational databases be developed. Non-relational databases, also called NoSQL databases, the most popular used databases are MongoDB, DocumentDB, Cassandra, HBase.

The phrase unstructured is data usually refers to information that doesn't reside in a traditional row-column database. As you might expect, it's the opposite of structured data the data stored in fields in a database. Unstructured data can be textual or non-textual. Textual unstructured data is generated in media like email messages, Power Point presentations, Word documents, collaboration software and instant messages. Non-textual unstructured data is generated in media like JPEG images, MP3 audio files and Flash video files.

Apache Flume is a tool for collecting and transporting large amounts of streaming data from various sources to a centralized data store. Spark fits into the Hadoop open-source community, building on top of the Hadoop Distributed File System (HDFS)[6]. However, Spark provides an easier to use alternative to Hadoop MapReduce and offers performance up to 10 times faster than previous generation systems like Hadoop MapReduce for certain applications. Fig 3 represents spark core components. In this spark streaming is required for our project.

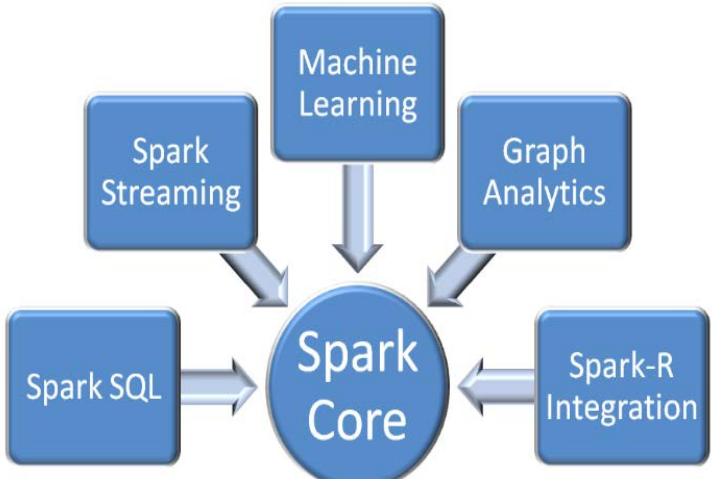

Fig 3: Spark Core Components

Spark Streaming is an extension of the core Spark API that enables scalable, high throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Twitter, Kinesis, and can be processed using complex algorithms expressed with high-level functions like map, reduce. Finally, processed data can be pushed out to file systems, databases, and live dashboards.

The rest of the paper is organized as follows. Section 2 describes related work; Section 3 represents the system model; Section 4 explains clear plan of work and the results; Section 5 concludes the paper with future work.

2. RELATED WORK

RDBMS(Relational Data Base Management System) is a database management system that is based on the relational model. RDBMS is structured in database like tables, fields and records. The limited amount of data will be present in RDBMS. Processing of large data is very difficult to handle and also time expenditure process. RDBMS cannot access semi structured and unstructured data. To overcome the drawbacks of this, Hadoop takes place. Hadoop is an open source java based programming framework that supports the processing and storage of extremely large datasets in a distributed computing environment. It can handle all types of data.

According to Ekata Banerjee[11],MongoDB is best for data storage,which is a NoSQL database. Its stored in the form of key value pairs which makes it easier for latter updation and creating the index than using traditional relational database.Now-a-days, unstructured data is growing in an explosive speed with the development of Internet and social networks.In my view MongoDB doesn't support all types of data. Instead of MongoDB here we are using the Cassandra. Cassandra supports all types of data in Hadoop 2.x version.

According to Elena Lazovik[12] Apache Storm is used for Analysis of Streaming Data. Here my proposesd tool for analyzing the streaming data is Apache Zeppelin. It is a web based notebook. Previously the concept of flume streaming is used for storing the streamed data into the hdfs. Now the concept of spark streaming is using for effective streaming.

According to Tripti Baraskar[14], HMR LOG ANALYZER is used for analyzing the data. In this map-reduce program is run.In our proposed system scala programming is run because scala is 100 times faster than the map reduce program. According to research and journals hadoop 2.x has high scalable than 1.x.In 1.x MR does both processing and cluster-resource management where as in 2.x YARN done cluster resource management and processing is done by using different processing models.

Our proposed system is using the following concepts for effective streaming data analysis.Apache flume, spark streaming, Cassandra and zeppelin. Apache flume is used to access the streaming data through twitter application. For any real-time data statistics, adaptive analysis and response, spark with spark streaming are a perfect fit. Flume is integrated with spark streaming and storing the data into Cassandra data storage. The twitter data is analyzed and validating by using apache zeppelin and json validation respectively. The result will be displayed on zeppelin dashboard.

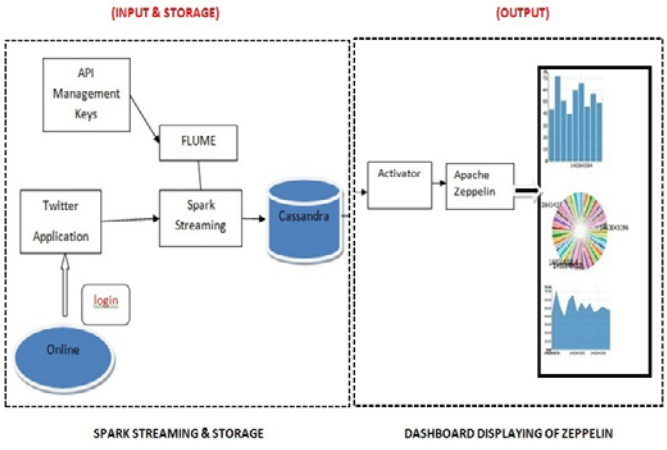

Fig 4: Architecture diagram for proposed system

The subsequent figure shows the architecture view for the proposed system. By this we can understand how our project is effective using the Hadoop and how the streamed data is going to be stored in the data storage from the Flume and spark streaming, also how it is going to display the result on dashboard.

3. SYSTEM MODEL

This section describes the overview of tools and technologies used in this project.

3.1 Extracting Twitter Data Using Flume:

Apache Flume is a tool for collecting and transporting large amounts of streaming data from various sources to a centralized data store. Using Flume, we can fetch data from various services and transport it to centralized stores.The following fig 5.describes how to extract the twitter data into hdfs.

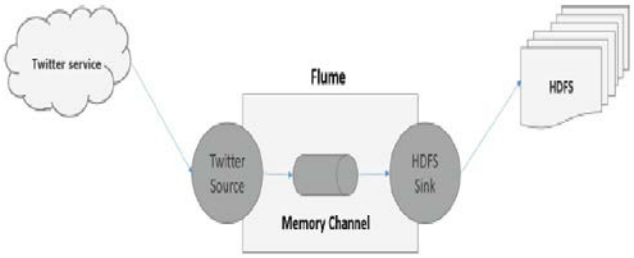

Fig 5:Apache flume streaming by using twitter source

In order to get the tweets from Twitter, it is needed to create a Twitter application. After creating the Application, the twitter agent will provide the consumer key & consumer secret key for accessing the tweets. As, the Twitter Streaming API gives a constant stream of tweet data coming from the application it must reside in HDFS securely. The security can be ensured by the generation of keys at the time of creating an application in twitter. This is called flume streaming.

3.2 Apache Flume is integrated with Spark Streaming.

Spark fits into the Hadoop open-source \ community, building on top of the Hadoop Distributed File System (HDFS). However, Spark provides an easier to use alternative to Hadoop MapReduce and offers performance up to 10 times faster than previous generation systems like Hadoop MapReduce for certain applications. Apache Flume is configured and integrated with spark streaming for streaming the data (unstructured) from twitter application. This is called spark streaming.

In most environments, Hadoop and Storm (or other stream processing systems) have been used for batch processing and stream processing, respectively. The use of two different programming models causes an increase in code size, number of bugs to fix, development effort, introduces a learning curve, and causes other issues. Spark Streaming helps in fixing these issues and provides a scalable, efficient and resilient.

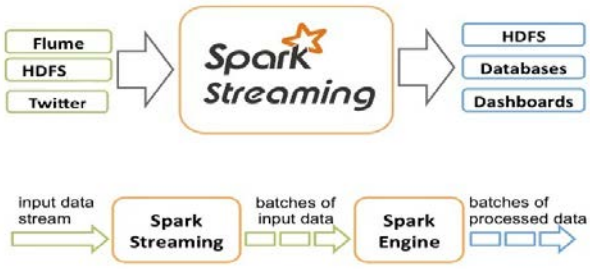

Fig 6: Describes spark streaming

In Spark Streaming, batches of Resilient Distributed Datasets (RDDs) are passed to Spark Streaming, which processes these batches using the Spark Engine as shown in the fig 6 and returns a processed stream of batches. The processed stream can be written to

a file system. The batch size can be as low as 0.5 seconds, leading to an end to end latency of less than 1 second. In this project scheduling algorithm is used.

3.3 Result will be displayed on dashboard by using Apache Zeppelin and JSON validator.

For the displaying of final results, first we need to store the streaming data into the Cassandra ***data storage***, after that analyzing the streamed data by using ***dashboard*** and validating the results using ***JSONLint.*** The following sections gives detailed flow.

Data Storage: Apache Cassandra is an open source, highly scalable and performance distributed database. It is designed to handle colossal of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL database. Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful "column family" data model. Cassandra is being used by some of the biggest companies such as Face book, Twitter, Cisco, eBay, and more. After retrieving the data from twitter, the data is stored in json format. Cassandra Referenced API class is the main entry point of the driver. It belongs to com.datastax.driver.core package. Session connect creates a new session on the current cluster and initializes it.

Apache Zeppelin: Apache Zeppelin is a web based and multipurpose notebook that enables interactive data analytics. The notebook is the place for data ingestion, discovery, analytics, visualization and collaboration. It can make beautiful data driven, interactive and collaborative documents with scala and more. Apache Zeppelin Interpreter allows any language/data-processing-backend to be plugged into Zeppelin. Currently Apache Zeppelin supports many interpreters such as Apache Spark, Python, JDBC, Markdown and Shell. Apache Zeppelin with Spark integration provides

- SparkContext and SQLContext injection.
- Runtime jar dependency loading from local file system.
- Canceling job and displaying its progress.

JSON Validation*:* JSON or JavaScript Object Notation is a web based tool and it is a language independent open data format that uses human readable text to express data objects consisting of attribute value pairs. JSON [15]validator is used to validate the json format data. The JSON Formatter was created to help with debugging. As JSON data is often output without line breaks to save space, it is extremely difficult to actually read and make sense of it. This little tool hoped to solve the problem by formatting the JSON data so that it is easy to read and debug by human beings.

4 PLAN OF WORK AND RESULTS

This project involves the following modules:

- Streaming the data from Twitter application- is the required input [Module I]
- Storing the streaming data into Cassandra [Module II]
- Streaming data analysis using Zeppelin and validating using JSON [Module III]

A.Streaming the data from Twitter application [Module I]

By the application of Twitter, JSON format files are generated which have to be dumped into the Cassandra. To fetch Twitter data, we will have to follow the steps given below-

- Create a twitter Application
- Install / Start HDFS→ Cassandra
- Configure Flume

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 10, October 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

- Spark streaming + flume integration

The following Fig 7 represents how to create a new twitter app by following the steps.

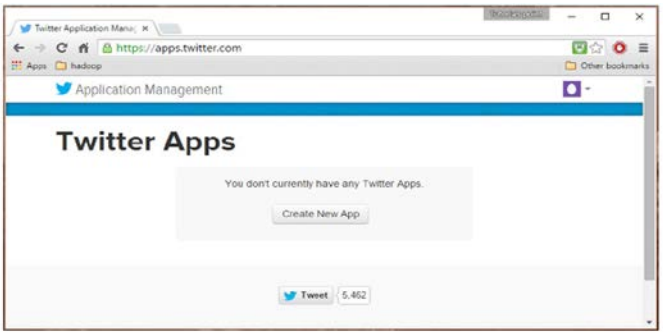

Fig 7: Create New Twitter App

To create a Twitter application, click on the following link https://apps.twitter.com/.Sign in to your Twitter account. You will have a Twitter Application Management window where you can create, delete, and manage Twitter Apps. Click on the Create New App button. You will be redirected to a window where you will get an application form in which you have to fill in your details in order to create the App. This will lead to a page which displays your Consumer key, Consumer secret, Access token, and Access token secret. Copy these details. These are useful to configure the agent in Flume. By following the steps mentioned above, the twitter app is created with given name. Fig 8 represents created the app by given name.

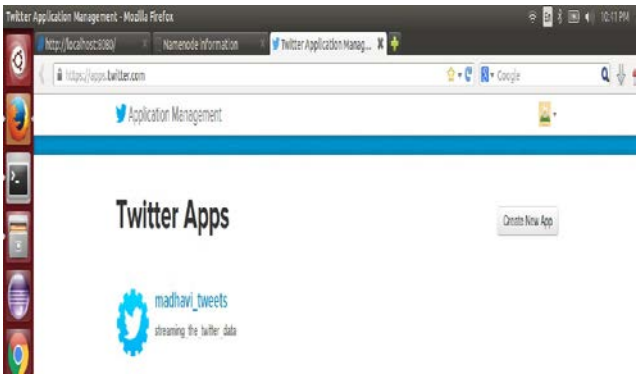

Fig 8:Created the Twitter App

There are two approaches to configure Flume and Spark Streaming to receive data from Flume: i) Flume-style Push-based Approach ii) Pull based Approach using a Custom Sink. Here we are using push based approach. For streaming the data first of all we need to login into our own twitter account.

B.Storing the streaming data into Cassandra [Module II]

The design goal of Cassandra is to handle big data workloads across multiple nodes without single point of failure. Cassandra has peer-to-peer distributed system across nodes, and data is distributed among all nodes in the cluster. Fig 9 represents how to integrate with hdfs. In this we have libraries.

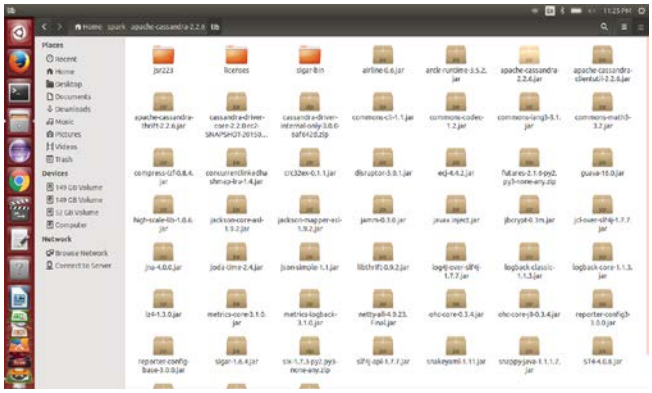

Fig 9: Apache Cassandra is integrated with HDFS

For Storing the streaming data into Cassandra[16] add the Cassandra-specific functions on the StreamingContext and RDD into scope, and simply replace the print to console with pipe the output to Cassandra:

import com.datastax.spark.connector.streaming._

wordCounts.saveToCassandra("streaming_test","words")

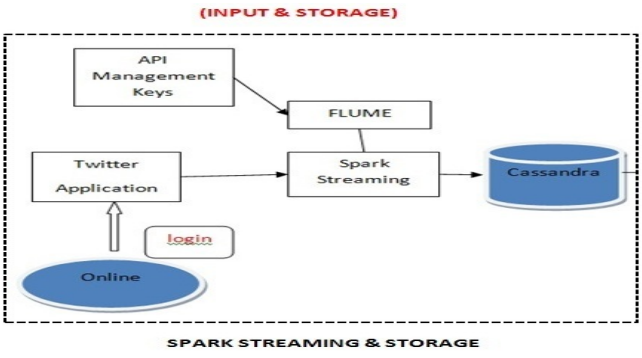

Fig 10:Spark Streaming & Storage

Fig 10 is an data flow diagram for input and storage. Here an input is streaming data and storage database is Cassandra. The main purpose of storing the streamed data is for analysis.

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
FlumeData.1470393894757	file	597.81 KB	1	128 MB	2016-08-05 16:48	rw-r--r--	hadoop	supergroup
FlumeData.1470395928079	file	621.90 KB	1	128 MB	2016-08-05 16:49	rw-r--r--	hadoop	supergroup
FlumeData.1470395959046	file	596.01 KB	1	128 MB	2016-08-05 16:49	rw-r--r--	hadoop	supergroup
FlumeData.1470395990013	file	7.67 MB	1	128 MB	2016-08-05 16:50	rw-r--r--	hadoop	supergroup
FlumeData.1470396021295	file	11.33 MB	1	128 MB	2016-08-05 16:50	rw-r--r--	hadoop	supergroup
FlumeData.1470396138363	file	608.01 KB	1	128 MB	2016-08-05 16:52	rw-r--r--	hadoop	supergroup
FlumeData.1470396172017	file	613.56 KB	1	128 MB	2016-08-05 16:53	rw-r--r--	hadoop	supergroup
FlumeData.1470396202974	file	593.58 KB	1	128 MB	2016-08-05 16:53	rw-r--r--	hadoop	supergroup
FlumeData.1470396233942	file	7.69 MB	1	128 MB	2016-08-05 16:54	rw-r--r--	hadoop	supergroup
FlumeData.1470396265174	file	11.36 MB	1	128 MB	2016-08-05 16:54	rw-r--r--	hadoop	supergroup

Fig 11:After Streaming the data stored in the web browser

After streaming the data, the data is going to be stored in web browser. To see this results in the web browser means go to the local host.

C.Streaming data analysis using Zeppelin and validating using JSON [Module III]

The main purpose of analyzing the streamed data is to know how many likes, shares and tweets etc., are coming for an images or videos and also in this we can easily identify who is the person posting or doing mentioned things from the huge amount of streamed data. Streaming data is basically a continuous group of data records generated from different sources. With streaming data processing, computing is done in real-time as data arrives rather than as a batch. Real-time data processing and analytics is becoming a critical component of the big data strategy for most organizations. Streaming data processing applications help with live dashboards, real-time online recommendations, and instant fraud detection.

After streaming the data, retrieving from apache Cassandra for analysis by using scala programming. The scala programming is configured on scala IDE 1.4.0 build on eclipse lona version. The scala IDE running the data by using Activator dist, these activator dist analyzes the streaming data. After analyzing the data the result will be displayed on apache zeppelin[18] dashboard.

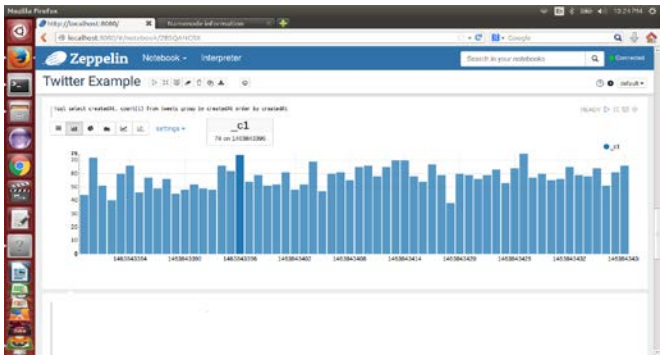

Fig 12: Dashboard result on bar chart

For the results, first we need to start the zeppelin, after that run the scala program. The results will be displayed as shown in the fig 12.

Fig 13: Dashboard result on pie chart

The dashboard will shows the results in different styles.Fig 13 represents results on pie chart. After displaying the results on dashboard,now validating the results.For this,first we need to select the customer_id which is shown in fig 12 for validating the results.When we clicked the particular id,the data is in unreadable format.By using the JSONLint we can read the data as shown in the figure.

Fig 14: JSON data validate by using JSONLint

The reading data contains the information like name,liked images or videos,tweets etc.,By this we can analyze the streamed data.

5 Conclusion

The generalized method called zeppelin for analyzing the streaming data with the help of apache flume, Cassandra and spark streaming. It's working on online applications like Twitter; it will access the streaming data. The data flow is maintained by the Apache spark streaming and dumped into the Cassandra data storage. The stored data or streaming data is analyzed and displayed on dashboard. In Hadoop 1.x name node failure will occur. The problem will cleared in Hadoop 2.x. When the data is storing the data loss will occur in HDFS where as in Cassandra no single point of failure. Spark streaming is efficient than flume streaming. After analyzing the data later we can do analytics. In future we can do this by using multiple nodes form in a cluster way.

REFERENCES:

[1] S.Vikram Phaneendra & E.Madhusudhan Reddy "Big Data-solutions for RDBMS problems- A survey" In 12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19{23 2013).

[2] Vavilapalli, A. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler, "Apache Hadoop YARN: Yet Another Resource Negotiator," in IEEE SOCC, pages 5:1–5:16, Santa Clara, CA, Oct. 2013.

[3] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1-12.

[4] Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N " Analysis of Bidgata using Apache Hadoop and Map Reduce" Volume 4, Issue 5, May 2014" 27.

[5] Kiran kumara Reddi & Dnvsl Indira "Different Technique to Transfer Big Data : survey" IEEE Transactions on 52(8) (Aug.2013) 2348 { 2355}

[6]HadoopDistributedFileSystem(HDFS), http://hortonworks.com/hadoop/hdfs/

[7] D.Fisher, R. DeLine, M.Czerwinsiki, and S. Drucker, "Interactions with big data analytics," Interactions, vol.19, no. 3, pp, 50_59, May 2012.

[8] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters," in HotCloud,2012.

[9] M. R. Ackermann, C. Lammersen, M. Märtens, C. Raupach,C. Sohler, and K. Swierkot, "StreamKM++: Aclustering algorithm for data streams," ACM Journal ofExperimental Algorithmics, vol. 17, 2012.

[10] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1-12.

[11] Jyoti Nandimath, Ankur Patil and Ekata Banerjee "Big Data Analysis Using Apache Hadoop" IEEE IRI 2013.

[12] Jan Sipke van der Veen, Bram van der Waaij, Elena Lazovik,"Dynamically Scaling Apache Storm for the Analysis of Streaming Data" IEEE 2015.

[13] Avita Katal, Mohammad Wazid AND R H Goudar "Big Data: Issues, Challenges, Tools and Good Practices", 978-1-4799-0192-0/13/$31.00 ©2013 IEEE.

[14]Sayalee Narkhede And Tripti Baraskar "Hmr Log Analyzer:Analyze Web Application Logs Over Hadoop Mapreduce" vol 4,no 3,july 2013.

[15] (Online Resource)http://jsonlint.com/

[16]ApacheCassandra "http://en.wikipedia.org/wiki/Apache_Cassandra.2014.

[17]"Designing performance monitoring tool for NoSQL Cassandradistributeddatabase". http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6360579&queryText%3DCASSANDRA, 2012 IEEE.

[18]ApacheZeppelin.Available: https://zeppelin.incubator.apache.org/