Abstract - In recent years, social media websites such as Twitter, Facebook, and Instagram have become very popular. Twitter, a popular social networking site is estimated to generate more than 200 million tweets a day and have 240+ million active users. In this paper, distributed computing platform offered by Cloudera is used to analyze real-time streaming Twitter data to find the sentiment expressed in each tweet. Sentiment mining is the process of determining the contextual polarity of text. In this paper, the paradigm to associate sentiments expressed by each tweet has been developed using Hadoop MapReduce framework. Also, the comparative analysis of Hadoop MapReduce and Apache Spark concerning the iterative computations to analyze their performance concerning memory usage and time has been performed. Performance of Apache Spark turned out to considerably higher almost 2x regarding time on a single node.

Keywords: Apache Spark, Big Data, Cloudera, Hadoop MapReduce, Sentiment Analysis, Twitter

1 Introduction

Social media websites such as Facebook, Instagram, WhatsApp and Twitter generate an enormous amount of data. The data produced by these social media websites are the short comment like status messages, including URL’s, hashtags, emoticons which are challenging to analyze. Besides, numerous applications require significant data processing which includes data about the environment, medical, smart grid, and social media. As the size of the data in such applications is very large, distributed processing is generally a necessity. Therefore, to process large distributed data, computing software platform such as Apache Hadoop [1] has emerged.

The paper is organized as follows. In section 2 we discuss prior works on sentiment analysis and Hadoop framework. In section 3 we present the Cloudera architecture. In section 4, we perform the comparative analysis between two data processing frameworks Hadoop MapReduce and Apache Spark. In section 5 we give details on collecting and storing tweets. In section 6, we present data preprocessing analyzing technique in MapReduce framework. In section 7, we conclude and give future references for our work in section 8.

2 Literature Review

The paper focuses on analyzing Twitter data using components of Cloudera Distribution of Hadoop (CDH) framework. It also discusses the comparison between the two major components of Hadoop framework, Apache Hadoop MapReduce, and Apache Spark. In this section, literature review based on previous research in this area has been presented.

Ying Zhou, Uwe Rohm and Bilal Akil (Dec 2017) [5] compares three most prominent distributed data processing platforms: Apache Hadoop MapReduce, Apache Spark, and Apache Flink, from a usability perspective. The focus is on data processing in the cloud, assessed with practical programming assignments. Stream processing is not covered in this usability study, all exercises in the paper are in the form of batch processing.

Juwei Shi, Yunjie Qiu and Chen Wang (Sept 2015) [6] dissects MapReduce and Spark frameworks and collect statistics from full-timers to quantify differences in their shuffle sort. They further conduct micro benchmark experiments like Word Count, Sort, k-means, and Page rank. They concluded Spark is approximately 2.5x, 5x and 5x faster than MapReduce, for Wordcount, k-means and Page Rank respectively. They further show that for iterative algorithms such as k-means and page rank, caching an input as RDDs can reduce both CPU and disk I/O overheads for subsequent iterations.

Satish Goplani and Rohan Arora (March 2015) [7] describes the difference between Apache Spark and Apache Hadoop MapReduce using K-Means cluster algorithm. It also shows the difference in Mapper and Reducer phase in Hadoop MapReduce and Spark.

Mahlakshmi R, Suseela (April 2015) [8] proposed a method of sentiment analysis and data visualization on twitter data using Hadoop and its components that process a large volume of data on a Hadoop, and the MapReduce function performs the sentiment analysis. Preprocessing of the data is not presented in the paper.
Michal Skuza and Andrzej Romanowski (Sept 2015) [9] explored the possibility of making a prediction of the stock market based on the classification of data coming from a Twitter microblogging platform. Unlike classical methods for forecasting macroeconomic quantities, prediction of future stock prices is performed here by combining results of sentiment classification of tweets and stock prices at different intervals.

Tare, Mohit, Indrajit Gahokar, and Jayant Sable (March 2014) [10] used Map-Reduce strategy for classification of tweets using Naïve Bayes classifier. In this, final reducer of map-reduce phase calculates the final probability of each category to which the tweet may belong to and outputs the predicted type and its probability.

Liu, Bingwei and Eric Blasch (Oct 2013) [11] worked on “Cornell university movie review dataset3” and implanted Naïve Bayes classifier to achieve excellent grain control of the analysis procedure for a Hadoop implementation. It resulted in 80.85% average accuracy.

3 Architecture

Apache Hadoop was created by Doug Cutting in 2005 when he was working with Yahoo. It is one of the open source framework integrated with big software company Cloudera known as CDH (Cloudera Distribution for Hadoop). It allows for the distributed processing of large data sets across clusters of computers using simple programming models. Within CDH ecosystem many other open source Apache frameworks allow us to express simple computation but hide the messy details of parallelization, fault tolerance, data distribution and load balancing such as Apache Hive, Apache Pig, and Apache Spark.

While multiple big data open-source frameworks are available in the market, choosing the suitable one for the problem is a challenge. Apache Hadoop has been leading the big data market for more than five years. Large processing of huge datasets can be achieved by the programming component of Apache Hadoop such as MapReduce, while Apache Spark delivers fast performance, real-time analytics, iterative processing, graph processing, machine learning and more.

Cloudera is a software company and sponsor of Apache Software Foundation that provides Apache Hadoop and Apache Spark-based software. Cloudera has created a functionally advanced system by integrating Apache Hadoop, i.e., CDH (Cloudera Distribution for Hadoop) and other major open source frameworks that help users to perform end-to-end big data workflows.

Figure 1 shows the complete architecture of the CDH. It provides a reliable, distributed platform to all Apache frameworks. With the advancement in CDH release, i.e., Hadoop version-2 in Oct 2014, Apache Spark also became an integral part of CDH.

4 Comparison between Hadoop MapReduce and Apache Spark

Before processing and analyzing data, comparative analysis of two frameworks Hadoop MapReduce and Apache Spark on a dummy dataset (.txt file) is performed. Dummy data is taken from HUCKLEBERRY FINN, by Mark Twain [5].

Three text files of sizes 15MB, 100MB, and 500MB are created on a single node of Cloudera Hadoop Architecture and computational time is monitored. Section A describes the procedure for performing the comparison to compute the time taken by these two frameworks for processing each file.

4.1 Procedure

First, the data has been preprocessed using regex function, followed by counting the occurrence of each word. Second, it is sorted by the order of occurrence. It implies two iterations are required to performing the work.

Figures 2 and 3 show the two output files of the experiment. Figure 2 depicts the occurrence of each word after the first iteration while Figure 2 depicts the result after the second iteration.

Section 4.2 and Section 4.3 discuss the implementation of the above procedure in Hadoop MapReduce and Apache Spark.
4.2 Hadoop MapReduce

In Hadoop MapReduce, the iterative operation is implemented through two MapReduce computations. In Hadoop MapReduce, every job reads its input data, process it and then writes it back to HDFS. So for every job it has to repeat read process and write cycle which makes the processing slow.

4.3 Apache Spark

In Spark platform, iterative operations are implemented in a single Spark program. There is no memory release phenomenon in Spark implementation. In Spark, data is stored on memory which shows a massive increase in job performance. In terms of writing, this processing framework requires about 90 percent fewer lines of code in comparison to that developed using MapReduce API.

Further Section 4.4 gives the configuration of the machine, required to perform the comparison, followed by section 4.5 in which result of comparative analysis between Hadoop MapReduce and Apache Spark are discussed.

4.4 Machine Configuration

- 8GB RAM
- Linux Ubuntu
- 20GB disk
- 2 CPU’s

4.5 Results

Execution time, or a duration time of the data processing task to complete, is one of the main criteria for evaluating the performance of each experiment using different data processing tools. Different input data will have an impact on execution time, in which larger size of data would cause a longer duration of processing time. The time duration of processing the data would also reduce when implementing a larger scale of cluster size since the additional number of nodes will help in processing the data in parallel. The result in Table 1 clearly shows that the performance of Apache Spark turned out to considerably higher regarding time on a single node.

<table>
<thead>
<tr>
<th>Dataset Size</th>
<th>Nodes</th>
<th>MapReduce Time(s)</th>
<th>Spark Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15MB</td>
<td>1</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>100MB</td>
<td>1</td>
<td>35</td>
<td>18</td>
</tr>
<tr>
<td>500MB</td>
<td>1</td>
<td>160</td>
<td>55</td>
</tr>
</tbody>
</table>

Apache Spark is the new software in Big Data area. Spark provides real-time, in-memory processing for those data sets that require it. When we combine, Apache Spark’s ability, i.e., high processing speed, advanced analytics and multiple integration support with Hadoop’s low-cost operation on commodity hardware, it gives the best results. However, regardless of whether Spark resembles the enormous framework, there is still a need to utilize the core components of Apache Hadoop like HDFS to store the information, and it might need to utilize HBase, Hive, Impala, or other Hadoop ventures. Our results on this research show that on following parameters, Spark is a robust framework. Hadoop MapReduce is a more mature platform and it was built for batch processing. It can be more cost-effective than Spark for truly Big Data that doesn’t fit in memory and also due to the greater availability.
5 Collecting Tweets

In this paper, Apache Flume is being used to collect data from Twitter Streaming API and forward it to HDFS. To access tweets, an open-source library in python called Tweepy is used. It provides access to the documented Twitter API. It supports accessing Twitter data through access token provided by the Outh2 library, which is the only way adopted by Twitter to secure its information.

Twitter Streaming API provided by Twitter allows the developer to access 1% of tweets based on the keyword tweeted at that time. All the Tweets have been collected in streaming fashion and therefore represent an accurate sample of actual tweets regarding the use of language and content [21]. Output file in HDFS is a streaming message and is represented as a JSON object, which stands for JavaScript Object Notation. It is a simple format for representing nested structures of data [22]. Figure 4 shows the structure of the single tweet in JSON format stored in HDFS. The JSON data will be a mix of root-level attributes, and child objects with the {} notation. A single tweet contains around 50 fields, e.g., screen_name, location, friends_count, retweets, followers_count, source and many more.

![Figure 4: Tweet Sample](image)

6 Data Processing and Analysis

After the tweets are extracted (sample shown in Figure 4), they are passed to the MapReduce framework for further processing and analysis. Figure 5 shows the flow diagram of a sample tweet in MapReduce framework. The MapReduce interface is based on two functional-programming primitives. Their signatures are reproduced here:

map: (kl, v1) → [(k2, v2)]
reduce: (k2, [v2]) → [v3]

The map function applies user-defined logic on every input key/value pair and transforms it into a list of intermediate key/value pairs. The reduce function applies user-defined logic to all intermediate values associated with the same intermediate key and produces a list of output values. This simplified interface enables developers to model their specific data processing into two-phase parallel tasks.

![Figure 5: Flow diagram of Tweets](image)

Mapper() and Reducer() functions are written to calculate the sentiment score of each tweet present in the output file. They describe a method to collect data sets of a keyword from the Twitter database using Twitter Streaming API. Tweets are then tokenized into tokens consisting of a single word. The polarity of each token is found using AFINN dictionary. The polarity ranges between [-5, 5]. Then the polarity of all the tokens in a tweet is summed up to assign a final polarity to a given tweet. Figure 6 shows the output of the reduce phase which consists of tweet number followed by the polarity of that tweet. Then each score/polarity is translated into emotion expressed in that tweet.

![Figure 6: Tweets Sentiment](image)
7 Conclusion

The focus of the paper is to assign the polarity of each tweet, i.e., whether the author expresses a positive or negative opinion. This paper gives the hands-on experience of dealing with massive amount of unstructured data in Hadoop Ecosystem. Table 2 shows the sentiment analysis of the tweets which were processed. Each score is translated into emotion expressed in that tweet.

<table>
<thead>
<tr>
<th>Tweets</th>
<th>Polarity</th>
<th>Emotion</th>
</tr>
</thead>
<tbody>
<tr>
<td>catch an early screening of the special May 25th at musicbox Chicago</td>
<td>1</td>
<td>Positive</td>
</tr>
<tr>
<td>(link below) or hang tight till June</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cambodia’s first martial arts movie is available now on listen to one</td>
<td>1</td>
<td>Positive</td>
</tr>
<tr>
<td>of the film’s stars</td>
<td></td>
<td></td>
</tr>
<tr>
<td>on netflix is shit crazy.</td>
<td>-4</td>
<td>Extremely negative</td>
</tr>
<tr>
<td>I think the bad not ill stick to watching it on Hulu or netflix</td>
<td>-1</td>
<td>Negative</td>
</tr>
</tbody>
</table>

The paper gave a learning opportunity about installation and configuration of the Hadoop distributed file system and the hands-on experience in writing algorithms in python using MapReduce paradigm. It provided the opportunity to do the comparative analysis of the two most powerful data processing framework in Hadoop Architecture, i.e., Hadoop MapReduce and Apache Spark. It also provided a paradigm to associate sentiment expressed in each tweet using Cloudera Distribution for Hadoop (CDH).

8 Future Work

The future of data analysis field is broad. This project can be extended to find other features such as friends/followers, retweets, and many more. Hadoop can be effectively used to compute the correlation between user influence and sentiment of the author. The emotional response of the user towards certain topic or product can be used for marketing. In the future, even richer linguistic analysis, for example, parsing, semantic analysis, and topic modeling can be explored.

9 References


