Comparison of MapReduce and Spark Programming Frameworks for Big Data Analytics on HDFS
Jai Prakash Verma¹, Atul Patel²
¹Assistant Professor, CSE Department, Institute of Technology, Nirma University, Ahmedabad, India
²Professor & Dean, CMPICA, CHARUSAT University, Changa, Gujarat, India

Abstract: Use of internet and all the types of computer automated systems generates large amount of data in different forms. Due to large volume, different types of varieties, and high velocity of this type of data emerges the Big Data Problem. Spark and MapReduce programming frameworks provide an effective open source solution for managing and analyzing the Big Data. Today researcher are comparing both the frameworks and making many interpretations which also generates many misconceptions about the performances and efficiency. In this paper we are discussing the working model of both the programming frameworks and by experimental analysis, we are also finding that Spark is three to four times faster than MapReduce paradigm on single node implementation of Hadoop Distributed File System.

Keywords: Big Data Analytics, Hadoop, Mahout, MapReduce, Spark Framework.

1. INTRODUCTION

Big Data cannot be a considered as a precise term with a proper definition. Instead, it is a characterization of the infinitely long accumulation of various kinds of data, mostly unstructured. This represents data which is too vast that relational database management systems won’t be able to analyses such data, because of the sheer size and unstructured nature. Any business should be capable of making use of data in the right way to create value on the table – value that can create better financial resources for the company or better experiences for the customers. Big Data Analytics provide a more precise solutions to researchers and analysts to use the previously unknown and unusable large amount of data available. Using advanced analytics techniques such as predictive analytics, text analytics, statistics, data mining, machine learning, and natural language processing, businesses can apply on these types of previously unknown and unused large dataset to find more valuable and useful insight which helps enterprises to make right decision as right time.

Open source technology like Hadoop/MapReduce and Spark provides an effective solution for Big Data Analytics[10, 11, 12]. In this paper we are discussing the comparison between Spark and MapReduce framework on Hadoop Distributed File System.

In in this paper Section –II covers related work done by different researchers to compare these two programming frameworks. As per environment setup and data size work presented in this paper is completely different from these papers. Section-III discusses the concepts of Big Data and Big Data Analytics. Section-IV covers open source solution Apache Hadoop with MapReduce Framework for storage and retrieval of large scale dataset. Section-V discusses about Mahout, the machine learning algorithm implemented in MapReduce programing concepts witch can run in distributed manner with Hadoop framework. Section-VI and VII discusses about Spark programming paradigm and MLlib library available with Spark for machine learning algorithms. Section –VIII produces the results and discussion generated by experimental study done with different size file with Spark and MapReduce programming framework on HDFS with single node Hadoop implementation.
2. RELATED WORK

Aaron N. Richter, Taghi M. Khoshgoftaar, Sara Landset, and Tawfiq Hasanin [5] proposed a complete multidimensional examination of different open source devices like Mahout, MLlib, H2O, and SAMOA for machine learning with huge information. An assessment standard is proposed alongside correlations of the structures talked about these open source technologies.


3. BIG DATA AND BIG DATA ANALYTICS

There are endless supply of both structured and unstructured information lying in the World Wide Web and this information could have answers to a considerable measure of inquiries, which haven't been encircled in the bothers of business pioneers. On the other hand these organizations don't figure out how to utilize Big Data to the best point of their interest, the open source technologies like Hadoop and Spark can be used to finding knowledge for the organizations. Organizations must discover powerful approaches to find knowledge from the huge amount of information being created each moment by making sense of what is in it and what to do with it. And there have been a few advancements in technologies which will add to the organizations making benefit out of Big Data.

These Big amount of data allude to five dimensions that are volume, velocity, variety, veracity and value. Volume: Enterprises are overwhelmed with regularly developing information of different kinds, effortlessly accumulating Exabyte—even Zettabytes—of data [1,2,3,4]. According to [4] 12 terabytes of Tweets made every day into improved product sentiment analysis, Convert 350 billion annual meter readings to better predict power consumption. Velocity: Data is increasing at unparalleled speed. We should think of what will happen in next coming years. It is a challenging task. Variety: As the increasing in technology, the data that comes today is in all type of formats. For example, audio, video, picture, numeric and structured data in traditional database. It is also becoming problem for the organizations nowadays.

Veracity: As per the different variety of data business analyst can trust on the information and knowledge generated. Because some time this types of data are not authentic as well as some time this types of data are sponsored or spam. As per [4] 1 in 3 business leaders don’t trust on the information or knowledge generated by computer automated systems for making decisions. Value: As value dimension researchers try to identify importance and usability of this huge amount of Big Data for enhancing business and living standard. Big amount of biometric data also follows these five dimensions, that why we can solve storing, managing, and retrieving issues using Big Data problem. Big data analytics, it is the process in which big data is examined that contains various kinds of data of various data types. The analysts find the hidden pattern in the big data and research the solution. The example of big data analytics done can be through text analysis, data mining etc. The steps involved in Big Data Analysis are: 1) Data Acquisition and Gathering 2) Data Cleaning and Extraction 3) Data Aggregation and Integration 4) Analysis and Modelling 5) Interpretation.

4. HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

Apache Hadoop/MapReduce is an open source solution for handling this large amount of data known as Big Data. Hadoop/MapReduce framework processes Big Data in parallel and in a fault tolerant manner. As per MapReduce programming paradigm a map/reduce job split the input data into number of memory chunks, and assign to different map tasks for processing completely in parallel manner. The MapReduce framework is responsible for sorting the output of map tasks and feed as input to the reduce tasks [1].

Apache Hadoop project includes following modules:

a) Hadoop Common: The common utility libraries which help to interact with other Hadoop components.
b) Hadoop Distributed File System (HDFS™): HDFS is a distributed file system which helps processes to access application data in high-throughput, in a fault tolerant and parallel manner.
c) Hadoop YARN: This framework is responsible for job scheduling and cluster resource management during a processes execution.
d) Hadoop MapReduce: It is a YARN-based system which responsible for parallel processing of large amount of data sets known as Big Data.

5. MAHOUT WITH MAPREDUCE FRAMEWORK

Apache Mahout is a machine learning and data mining framework for classification, clustering, Frequent Itemset and recommendation which can run on Hadoop/MapReduce framework in distributed manner. It is an open source, scalable machine learning libraries which can be used for analyzing big data on a distributed manner. Apache Mahout...
For experimental analysis we are running wordcount program with different sizes files on Spark and MarpReduce Framework. Here we are running both the open source tools on single node Hadoop installation on Ubuntu machine.

A. Software includes three major features: First: Mahout is a simple and extensible programming environment and framework for building scalable algorithms. Second: Mahout provides a wide variety of premade algorithms comparative to Scala + Apache Spark, H2O, and Apache Flink. Third: Mahout is an established set of libraries which implemented classification, clustering, Frequent Itemset and recommendation algorithms that can run on Hadoop MapReduce environment.

6. SPARK FRAMEWORK AND ARCHITECTURE

Apache Spark (figure-1) is an alternative open source solution for Big Data Analytics which can be work in distributed manner in the way of data stoppage as well as analysis. Apache spark introduce RDD (Resilient Distributed Dataset) which provides an application programming interface centered on a data structure. RDD is an immutable fault-tolerant, distributed collection of objects that can be operated on in parallel. In MapReduce cluster computing paradigm each map and reduce phase disk read and write operation were performed which causes extra delay in execution. Spark's RDDs provides the functionality where programmer can write a program in a distributed manner that offers a (deliberately) restricted form of distributed shared memory because RDD helps to persist the data in RAM and help for effectively process these data. It is the technology that efficiently utilize in-memory LRU cache with possible on-disk eviction on memory full condition. And it puts all the dataset data on the local file systems during the “shuffle” process [13,14].

Figure 1:- Spark Architecture

7. MACHINE LEARNING LIBRARIES (MLlib) WITH SPARK

Spark additionally includes libraries for doing things like machine learning, streaming, graph programming and SQL (see the figure -1). This additionally included libraries makes things much simpler for programmer and developer. These libraries are incorporated, so many enhancements in Spark as per time period which gives advantages to the additional packages as well. Most data analyst would somehow or another need to fall back on utilizing heaps of other random tools to complete their work, which makes things complex. Spark’s libraries are intended to all work together, on the same bit of information, which is more coordinated and less demanding to utilize. Spark streaming specifically gives an approach to do ongoing stream handling. The previously stated. Spark will empower programmer to do real-time analysis of everything from data available with an enterprise. The data may be trading data or web clicks, Spark provide an ease and faster open source solution to handle the Big Data problem for an enterprise [13,14].

8. EXPERIMENTAL CASE STUDY

For experimental analysis we are running wordcount program with different sizes files on Spark and MapReduce Framework. Here we are running both the open source tools on single node Hadoop installation on Ubuntu machine.

A. Dataset Selection: Here we are using a large text file that represents customer review and feedback about the different products. We split the same file in different sizes for performance analysis.

B. Methodology and Algorithm: Here we executed wordcount program written on MapReduce programming paradigm and another wordcounkt program written on spark programming paradigm. Both the open source tools were running on single node Hadoop Distributed File System implemented with Ubuntu operating system.

C. Findings:Table - 1 shows the CPU execution time for wordcount program on both the programming frameworks. Figure - 1 shows the comparison of spark and mapreduce program execution. Here the graph shows that Spark is three to four times faster than mapreduce framework. Figure – 2 shows the graph between CPU execution time for both the tools with different sizes files.
TABLE I. PERFORMANCE EVALUATION FOR WORDCOUNT PROGRAM

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Text File Size (in Bytes)</th>
<th>MapReduce Framework</th>
<th>Spark Framework</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>153.3 kB</td>
<td>156979.2</td>
<td>3.15</td>
<td>1.053078</td>
</tr>
<tr>
<td>2</td>
<td>260.1 kB</td>
<td>266342.4</td>
<td>3.67</td>
<td>1.110998</td>
</tr>
<tr>
<td>3</td>
<td>358.4 kB</td>
<td>367001.6</td>
<td>3.52</td>
<td>1.132921</td>
</tr>
<tr>
<td>4</td>
<td>769.3 kB</td>
<td>787763.2</td>
<td>4.5</td>
<td>1.240024</td>
</tr>
<tr>
<td>5</td>
<td>1.0 MB</td>
<td>1048576</td>
<td>5.06</td>
<td>1.38958</td>
</tr>
<tr>
<td>6</td>
<td>2.8 MB</td>
<td>2936012.8</td>
<td>7.08</td>
<td>1.770073</td>
</tr>
<tr>
<td>7</td>
<td>5.1 MB</td>
<td>5347737.6</td>
<td>8.83</td>
<td>2.123277</td>
</tr>
<tr>
<td>8</td>
<td>10.5 MB</td>
<td>11010048</td>
<td>10.46</td>
<td>3.132935</td>
</tr>
<tr>
<td>9</td>
<td>15.6 MB</td>
<td>16357785.6</td>
<td>12.5</td>
<td>4.874238</td>
</tr>
<tr>
<td>10</td>
<td>26.1 MB</td>
<td>27367833.6</td>
<td>16.81</td>
<td>5.329568</td>
</tr>
<tr>
<td>11</td>
<td>55.1 MB</td>
<td>57776537.6</td>
<td>26.83</td>
<td>9.652699</td>
</tr>
<tr>
<td>12</td>
<td>100.3 MB</td>
<td>105172173</td>
<td>54.86</td>
<td>20.168503</td>
</tr>
<tr>
<td>13</td>
<td>255.9 MB</td>
<td>268330598</td>
<td>131.46</td>
<td>33.377112</td>
</tr>
<tr>
<td>14</td>
<td>329.3 MB</td>
<td>345296077</td>
<td>161.76</td>
<td>49.030878</td>
</tr>
<tr>
<td>15</td>
<td>1.2 GB</td>
<td>1288490189</td>
<td>610.65</td>
<td>147.019231</td>
</tr>
<tr>
<td>16</td>
<td>1.6 GB</td>
<td>1717986918</td>
<td>791.09</td>
<td>187.673476</td>
</tr>
<tr>
<td>17</td>
<td>2.8 GB</td>
<td>3006477107</td>
<td>1395.84</td>
<td>337.269451</td>
</tr>
<tr>
<td>18</td>
<td>4.7 GB</td>
<td>5046586573</td>
<td>1731.95</td>
<td>550.205804</td>
</tr>
<tr>
<td>19</td>
<td>7.5 GB</td>
<td>8053063680</td>
<td>3604.45</td>
<td>835.431703</td>
</tr>
</tbody>
</table>

Figure 2: Comparison of Wordcount program execution between Spark and MapReduce Framework

Figure 3: Comparison of MapReduce and Spark Wordcount Program Execution
D. Discussions:

As per architecture and workflow of Spark and MapReduce frameworks, Spark is much faster than MapReduce because it perform IO (Input/Output) operation for read and write data on HDD once at the times of shuffles but MapReduce programming model perform at the time of every map and reduce task. In MapReduce workflow there are series of map and reduce jobs, each of which perform data read and write operation to HDD between iterations. With the supports of DAGs and pipelining Spark perform the complex workflows without intermediate data materialization (unless you need to “shuffle” it) caching. In general the concept of in memory execution enhance the performance of Spark over Hadoop/MapReduce framework. Here the meaning of in memory execution is caching the read and write operation without interaction with HDD. As per experimental study done with Spark on single node Hadoop implementation in section-8 shows Spark can be executed 3 to 4 times faster than MapReduce programming framework (Figure -2).

The key difference between Spark and MapReduce programming concept is spark’s in memory execution of read and write operation require as the time of shuffling of data. For every MapReduce job data are read from HDFS and write to HDFS in each data shuffle iteration but in spark it performs once.

9. CONCLUSION

Spark and MapReduce framework both are popular distributed computing paradigm for providing an effective solution for handling this large amount of data called Big Data. Today there are many misconception about the comparison of spark and mapreduce framework. Many researchers and bloggers are claiming that Spark is 10 to 100 times faster than mapreduce framework. In this paper we are also comparing both the programming frameworks. We have run workcount program on both the programming framework with different size of files and found that Spark is 3 to 4 times faster in single node Hadoop implementation.

REFERENCES


DOI: 10.090592/IJCSC.2016.113