International Journal of
Scientific Engineering and Technology Research

www.ijsetr.com

ISSN 2319-8885
Vol.07,Issue.07,
July-2018,
Pages:1361-1364

Weather Data Analytics using Apache Spark

VAKA PRAVEENKUMAR REDDY[1], G. LAKSHMI VARA PRASAD[2]
[1]PG Scholar, Dept of CSE, QIS College of Engineering & Technology, Ongole, AP, India.
[2]Associate Professor, Dept of CSE, QIS College of Engineering & Technology, Ongole, AP, India.

Abstract: Weather forecasting is a motivational challenge in human civilization. There is a shift in scientific focus towards solving this problem. Weather is a complex phenomenon that involves dynamic interaction of several forces. There are number of numerical weather models and algorithms that have been developed and enforced to predict the weather forecasting. Weather forecasting plays a very crucial role in various fields like agriculture, government planning, industries, predicting stock analysis etc. Various sensors are deployed at different geographical locations to collect weather data on a daily basis. The greatest challenge is to store and analyze the huge volume of data in an effective manner. Big data technologies are used to address these issues and challenges using distributed computing. HadoopMapReduce is used to analyze the weather data. But spark is an emerging technology that performs in-memory computing which is more efficient than Hadoopmapreduce. This paper proposes the spark implementation using Scala Programing Language for weather data analysis and shows the highest average precipitation, temperature or any weather related parameters for different weather stations.

Keywords: Weather Forecasting, Hadoop, MapReduce, Spark and Scala.

I. INTRODUCTION

Big data analytics plays an important role in managing very large amounts of data and extract value and knowledge from them. The different challenges of big data are scalability, complexity and speed. Scaling refers to data volume which is increasing exponentially from Terabytes to Petabytes, Exabytes, Zettabytes and Yottabytes. Complexity refers to data variety which can be structured, unstructured and semi-structured data. It includes various formats, types, and structure like Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc. Speed refers to the variety where the data is being generated fast and need to be processed fast. Examples include Online Data Analytics, e-promotions and healthcare monitoring where the sensors will monitor your activities and any abnormal measurements require immediate reaction. In the upcoming years, the data growth will be still increasing exponentially and there is a need to find the efficient methodologies for real-time processing of big data and information extraction. Every country has a meteorological department. It possesses the data which is collected from various sensors for weather parameters. At each location the values of various weather parameters is collected at a frequency of 3-4 times per hour. This data is stored in the unstructured format along with location, date and time. Direct processing of this huge unstructured data using conventional methods and tools is difficult and inefficient. This has resulted in the challenges of storage and processing of enormous weather data. One of such data is stored at NCDC, USA. It has the repository for weather data from last many years till today.

The structure of these formats is flat file which is separated by comma or tab or may be semicolons. It is difficult to process this unstructured data directly. The collective data is becoming very huge considering various parameters, their frequency of recording and number of locations. Day by day this data is growing and accumulated at enormous speed. Hence, to process this data using conventional methods and tools is becoming a challenge. Hadoop platform with MapReduce programming paradigm has proven to be very useful in processing huge unstructured data. Spark with in memory computing also gives very good performance then compared to MapReduce for analysis of unstructured data.

II. LITERATURE SURVEY

Basvanth Reddy and Prof. B.A.Patil [3] worked on Weather Prediction Based on Big Data Using Hadoop Map Reduce Technique and find a temperature of a particular city for a particular year. The weather data was taken from NCDC.

Doreswamy, Gad Ibrahim [4] worked on building a platform using Hadoop to analyse the weather data. Temperature and yearly precipitation were chosen as weather parameter for extraction and analysis.

Y. VenkataRaghavarao, L. S. S Reddy [8] worked on the different data sources like HDFS, Kafka, and Flume are used to ingest the data or new custom data sources can also be created. [6] [8] Dstreams are sequence of Resilient Distributed Datasets (RDDs) where each RDD contains the records of batch processing.

III. DISCUSSION

Apache Spark is an open-source cluster computing framework for real-time processing. It has a thriving open-source community and is the most active Apache project at the moment. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance as shown in Fig.1.

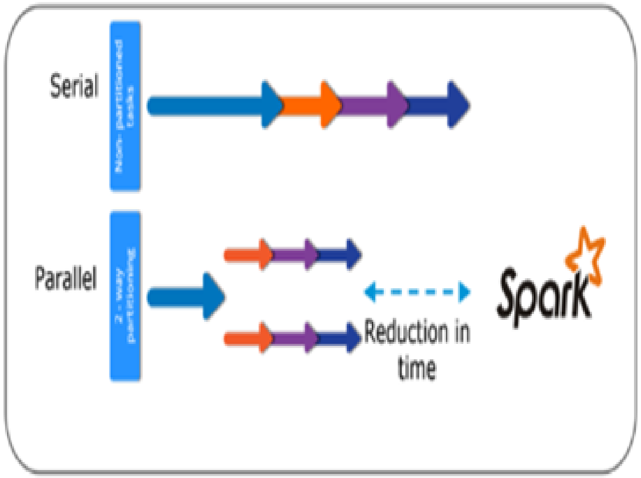

Fig.1. Real time processing in Apache Spark.

It was built on top of HadoopMapReduce and it extends the MapReduce model to efficiently use more types of computations.

A. Spark Has The Following Features

Polyglot: Spark provides high-level APIs in Java, Scala, Python and R. Spark code can be written in any of these four languages. It provides a shell in Scala and Python. The Scala shell can be accessed through ./bin/spark-shell and Python shell through ./bin/pyspark from the installed directory.

Speed: Spark runs up to 100 times faster than HadoopMapReduce for large-scale data processing. Spark is able to achieve this speed through controlled partitioning. It manages data using partitions that help parallelize distributed data processing with minimal network traffic.

Multiple Formats: Spark supports multiple data sources such as Parquet, JSON, Hive and Cassandra apart from the usual formats such as text files, CSV and RDBMS tables. The Data Source API provides a pluggable mechanism for accessing structured data though Spark SQL. Data sources can be more than just simple pipes that convert data and pull it into Spark.

Lazy Evaluation: Apache Spark delays its evaluation till it is absolutely necessary. This is one of the key factors contributing to its speed. For transformations, Spark adds them to a DAG (Directed Acyclic Graph) of computation and only when the driver requests some data, does this DAG actually gets executed.

Real Time Computation: Spark's computation is real-time and has low latency because of its in-memory computation. Spark is designed for massive scalability and the Spark team has documented users of the system running production clusters with thousands of nodes and supports several computational models.

RDD: Resilient Distributed Dataset (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

Hadoop Integration: Apache Spark provides smooth compatibility with Hadoop. This is a boon for all the Big Data engineers who started their careers with Hadoop. Spark is a potential replacement for the MapReduce functions of Hadoop, while Spark has the ability to run on top of an existing Hadoop cluster using YARN for resource scheduling.

B. Components of Spark

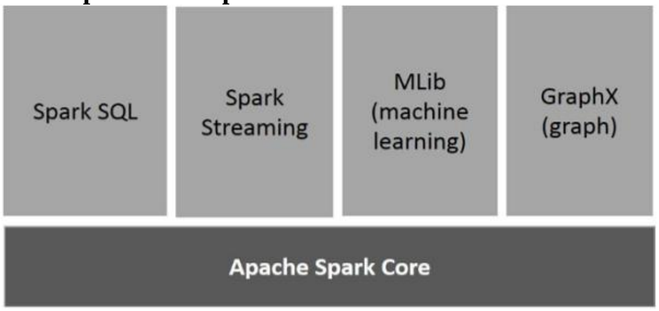

Fig.2. Components of Spark.

Apache Spark Core: Spark Core is the underlying general execution engine for spark platform that all other functionality is built upon as shown in Fig.2. It provides In-Memory computing and referencing datasets in external storage systems.

Spark SQL: Spark SQL is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.

Spark Streaming: Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD (Resilient Distributed Datasets) transformations on those mini-batches of data.

MLlib (Machine Learning Library): MLlib is a distributed machine learning framework above Spark because of the distributed memory-based Spark architecture. It is, according to benchmarks, done by the MLlib developers against the Alternating Least Squares (ALS) implementations. Spark MLlib is nine times as fast as the Hadoop disk-based version of Apache Mahout (before Mahout gained a Spark interface).

GraphX: GraphX is a distributed graph-processing framework on top of Spark. It provides an API for expressing graph computation that can model the user-defined graphs by using Pregel abstraction API. It also provides an optimized runtime for this abstraction.

IV. METHODOLOHY

A. Input Dataset

In this project, the algorithm is implemented for calculation of minimum, maximum and average values of temperature, pressure. Data is collected from NCDC web site from years 1996 to years 2018. NCDC has data for each month with hourly basis with a frequency of 3-4 records per hour. The data has fields for time, date, location id and observations for each weather parameters viz... temperature, pressure, humidity and wind speed and some other parameters also.

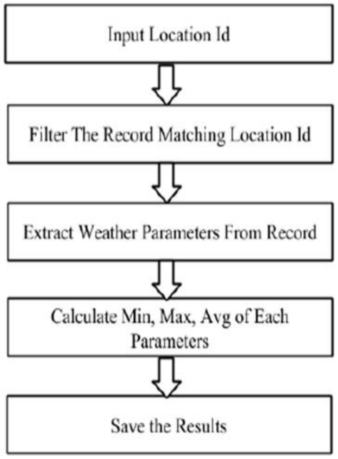
Input Location Id
Filter The Record Matching Location Id
Extract Weather Parameters From Record
Calculate Min, Max, Avg of Each Parameters
Save the Results

Fig.3. General Algorithm Flowchart for Weather Data Analysis.

B. General Algorithm Flowchart

The general algorithm flowchart is shown in Fig.3. Each weather record is filtered for a particular location id which is input to the tool. From the filtered in record the values of weather parameters are extracted. After that values of each parameter for a particular day is aggregated and their minimum, maximum and average is calculated.

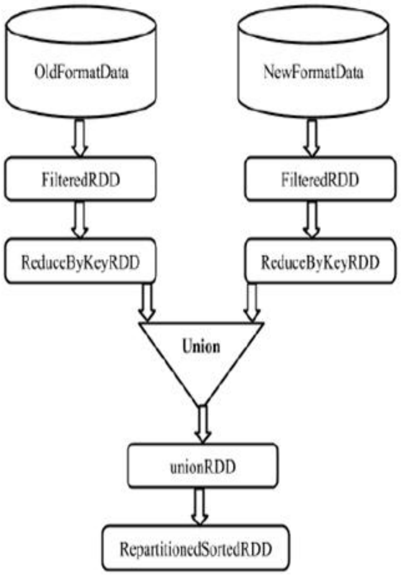

Fig.5. Algorithm Flowchart for Weather Data Analysis Using Spark Implementation.

The general algorithm flowchart for weather data analytics using Spark is shown in Fig.4. Weather data which is in old format and new format is stored on HDFS. The location id is given as input to the program. The Spark program generally is the sequence of RDD transformations. The FilteredRDD filters the record matching the location id. It also extracts the temperature, pressure, humidity and wind speed as weather parameters. The ReduceByKeyRDD transformation calculates the minimum, maximum and average values of each weather parameters. The union operation merges the two ReduceByKeyRDD to make a single UnionRDD. The RepartitionedAndSortedRDD repartitions and sorts the output before results is stored back to the HDFS. The repartitioning is necessary so that the related records must come to the same block before sorting is applied. The union operation is used from Spark framework which is already implemented in an optimized manner. The whole algorithm is composed of RDD transformations. This is recommended approach for Spark program. It results in lot of in memory computation leading to an optimized performance.

V. CONCLUSION

The meteorological department from each country collects huge amount of weather data which is being generated every day. This has resulted in the challenge of storing and processing of enormous weather data. In this study important weather parameter like temperature, pressure, humidity and wind speed are analyzed by calculating the minimum, maximum and average values of each parameter. The weather analysis is done using Spark. The benchmarking of weather data shows that the performance of Spark is very much better than the MapReduce implementation. In the proposed implementation of Spark has 1.5 x performance than the MapReduce implementation for weather data analytics. Hence, it can be concluded that the Spark has better performance for weather data analytics than the MapReduce.

VI. ACKNOWLEDGMENT

I would like to acknowledge my Principal at College, QIS College of Engineering & Technology (QISCET) Ongole, India and the staff of college and my friends for supporting and motivating me for my research work. I would like to thank my family members for their support.

VII. REFERENCES

[1] "Big Data Analytics in the Cloud: Spark on Hadoopvs MPI/OpenMP on Beowulf", 2015 INNS Conference on Big Data, vol. 53, pp. 121-130, 2015.

[2] "An Evaluation of Data Stream Processing Systems for Data Driven Applications", ICCS 2016. The International Conference on Computational Science, vol. 80, pp. 439-449, 2016.

[3] BasvanthReddyl, B.A Patil, "Weather Prediction Based on Big Data Using Hadoop Map Reduce Technique", International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 6, June 2016, ISBN 2278-1021.

[4] Doreswamy, Gad Ibrahim, "Big data techniques:hadoop and mapreduce for weather forecasting", International

Journal of Latest Trends in Engineering and Technology, pp. 194-199, 2016.

[5] Rahul Palamattom, Renato Marroquin, Chris Mattmann, "Sci-Spark: Applying In-memory computing framework to weather event detection and tracking", 2015 IEEE International Conference on Big Data (Big Data) 978-1-4799-9926-2/15/\$31.00©2015 IEEE, pp. 2020-2026.

[6] Choi Dojin, Song Seokil, "Processing Moving Objects and Traffic Events based on Spark Streaming", 2015 8th International Conference on Disaster Recovery and Business Continuity 978-1-4673-9840-4/15 \$31.00 © 2015 IEEE.

[7] AlttiIlariMaarala, Mika Rautiainen, "Low latency analytics for streaming traffic data with Apache Spark", 2015 IEEE International Conference on Big Data (Big Data) 978–1-4799-9926-2/15/\$31.00 ©2015 IEEE.

[8] Y. VenkataRaghavarao, L. S. S Reddy, "Map Reducing Stream Based Apriori in Distributed Big Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 7, July 2014.

Author’s Profile:

Mr. Vaka Praveenkumar Reddy M.Tech Scholar in Computer Science and Engineering, at QIS College of Engineering and Technology (QISCET), Ongole, India. He has done B.Tech in Electronics and Communication Engineering from Rise Prakasam Group of Institutions, Ongole, India. His area of research is in Parallel and Distributed Computing, and Big Data Analytics.

Mr. G Lakshmi Vara Prasad has received B.E from Anna University and M.Tech from Bharath Institute of Higher Education and Research (BIHER). He is pursuing Ph.D. in BIHER, Chennai. He is dedicated to teaching field from the last 8 years. His research areas included Big Data Analytics. At present he is working as Associate Professor in QIS College of Engineering & Technology (Autonomous), Ongole, Andhra Pradesh, India.