ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Weather Data Analytics using MapReduce and Spark

Priyanka Chouksey[1], Abhishek Singh Chauhan[2]

M.Tech Research Scholar, NIIST, Bhopal (M.P), India[1]

Assistant Professor, NIIST, Bhopal (M.P), India[2]

Abstract: Weather data analytics is very important in every aspect of human life. Weather plays a crucial role in every sectors like agriculture, tourism, government planning, industry and many more. Weather has various parameters like temperature, pressure, humidity and wind speed. The meteorological department from every country has deployed sensors for each weather parameter at various geographical locations. From these sensors weather data is collected on a daily basis. This data is stored mostly in the unstructured format. Due to this, huge amount of data has been collected and archived. Hence, storage and processing of this data for accurate weather prediction is a big challenge. Big data technology like Hadoop and Spark have evolved to solve the challenges and issues of big data using distributed computing. Till date few studies have been reported on the processing of weather data using MapReduce. Similarly, Spark which is the emerging technology claims to be in-memory computing can be applied for weather data analytics. This project presents the analysis of weather data by calculating minimum, maximum and average values of weather parameters. The code is implemented in both MapReduce and Spark to study their relative performance for the weather data analytics.

Keywords: Big Data, Hadoop, Spark, MapReduce, Weather Data Analytics.

I. INTRODUCTION

Weather has a direct impact on human life. Every individual is directly or indirectly affected by weather. Due to this weather data analytics is a crucial domain for study. Agriculture sector is most dependent sector on weather prediction. Similarly tourism sector get affected by weather. Lots of government bodies are interested in weather data analysis for their strategic planning in case of flood and drought. Human mood and health can also be affected by weather [1]. Weather forecasting can result is advance preparation for any natural calamities. So across globe, there is a lot of interest in weather data analysis which may be in the form for global warming, temperature prediction, precipitation and many more.

Almost every country has a meteorological department which does weather prediction. It possesses the data which is collected from various sensors for weather parameters [2]. Some of the crucial parameters are temperature, pressure, humidity and wind speed. The data is collected on hourly basis with a frequency of 3-4 times per hour. Typically, this data is stored in the unstructured format. The structure of this data format is a plain text file where each field is separated by a comma or a tab or may be by the semicolon. This huge amount of data has got accumulated from last many years and it will continue to grow. 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 [3]. It has the repository for weather data from last many years till today.

Various technologies like Hadoop [5], Spark [6], Storm, NoSQL [7] has evolved to address the challenges of Big Data [4]. Out of this technology Hadoop MapReduce which is efficient for batch processing and Spark which is efficient for iterative in-memory computing [8] are the most prominent. It is important to study their relative performance and usefulness in various domains.

In the current project, the weather data analytics [9] is done by calculation of minimum, maximum and average values of each weather parameters viz. temperature, pressure, humidity and wind speed. The code for analysis is implemented using both MapReduce and Spark. The benchmarking was compared between these two methods on datasets of various sizes.

II. DISCUSSION

Data is growing at a huge scale with an enormous speed. Various domains like Share market, Social media, Bioinformatics, Scientific experiments and Meteorological departments are producing huge amount of data. The rate of growth of these data is increasing every year at least by two fold. These domains are generating data in mainly three types viz. structured, semi structured and unstructured data.

There are various tools and techniques available to store and process structured and semi structured data efficiently. But, storage and processing of the unstructured data which constitutes more than 80% of total data is a major

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

challenge. Because, processing the unstructured data directly using conventional tools and methods is difficult and inefficient. Hence, there is a need of special tools and techniques to address the challenges of storage and processing of such huge unstructured Big Data.

Big Data has three major characteristics as explained below. Apache Hadoop and Spark are the few technology which addresses the challenges of Big Data [4].

Volume - It means the huge size of data which has been generated and got accumulated. Different meteorological departments have the archival of huge weather data which is being collected from many years. Storage and Processing of this huge data is a challenge.

Velocity - It refers to the pace at which data is being generated. Various sensors and IoT devices are generating huge amount of data per unit time. Conventional data processing tools are not able to process data at the same or equivalent rate. This has led to the huge amount of unprocessed data accumulation and delayed analysis.

Variety - It refers to various formats in which data is being generated from various sources. Like, unstructured text data, multimedia data like images, audio and video data. Even in text format data may be in plain ASCII, XML or JSON format. Processing such huge variety of data is a big challenge.

A. Hadoop

The Apache Hadoop software library is a framework implemented in java that allows for the distributed processing of large data sets across clusters of commodity computers using simple programming models [5]. It is designed to scale up from single nodes to thousands of nodes. Each node offers local computation and storage. Hadoop does not depend on hardware to deliver high-availability. The Hadoop the library itself is designed to detect and handle failures at the application layer. Hadoop delivers a highly-available service on top of a cluster of commodity computers. Each commodity computer may be prone to failures.

Hadoop is good for batch processing job. Hadoop has three major components viz. Hadoop Distributed File System (HDFS), MapReduce Processing Engine and YARN as shown in figure 1. On top of Hadoop other programming framework like Spark can also run.

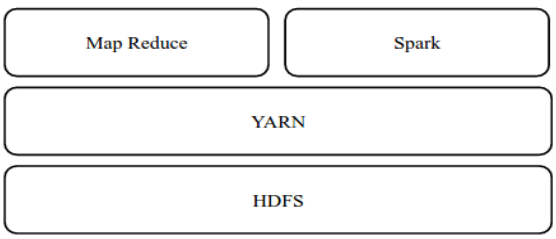

Figure 1 - General Layered Architecture of Hadoop

HDFS - It is the distributed file system derived the idea from the Google file system published in white paper [10]. It is the primary distributed storage used by the Hadoop applications. It is block oriented, fault tolerant, scalable and distributed file system. It supports huge amount of storage which can give streaming access to a data. HDFS has master slave architecture. It has a namenode which acts as master and multiple datanode which act as storage and computation unit. Data is stored in datanode with 128MB block size.

MapReduce - MapReduce is the programming paradigm for processing huge amount of data stored on large clusters of commodity hardware in a parallel and distributed manner [11]. In MapReduce paradigm the business logic is encoded in the key value pair format. Key value pair is the core of MapReduce programming. It has Mapper and Reducer along with intermediate combiner, partitioner, shuffle and sort phase which can be overwritten to customize their default behavior.

YARN - YARN is Yet Another Resource Negotiator. It was added to Hadoop framework from Hadoop 2.x [5]. In Hadoop 1.x the MapReduce part had the responsibility of both processing and cluster management. YARN takes the responsibility of resource management from MapReduce engine. It gives the separation between job execution and resource management. It also enables the platform to run another type of programing model like Spark, Storm, and Apex to be executed on top of it.

B. Spark

Apache spark is fast and general purpose engine for large scale data processing [6]. Architecture of apache spark is shown in figure 2. It has spark core at its bottom and on top of which Spark SQL, Spark streaming, Spark MLlib and Spark GraphX libraries are provided for data processing. Apache Spark provides in memory computing which avoids latency and given good performance. Spark has Mesos as its own cluster management but it can work with Hadoop YARN cluster also.

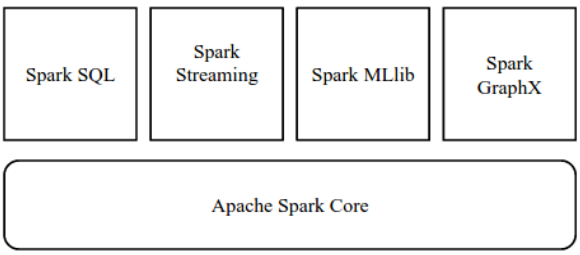

Figure 2 - General Layered Architecture of Hadoop

There are three core building blocks of Spark programming. Resilient Distributed Datasets (RDD), Transformations and Action.

An RDD is immutable distributed collection of objects. Distributed means, each RDD is divided into multiple partitions [8]. Each of these partitions can reside in

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

memory or stored on disk of different machines in a cluster. On RDD either transformation can be executed or any action is called.

Transformations create new RDD from existing RDD as original RDD is immutable. RDDs maintain a graph of one RDD getting transformed into another called lineage graph, which helps Spark to recompute any intermediate RDD in case of failures. This way spark achieves fault tolerance. This is because transformations are executed on demand and they are computed lazily. Actions return final results of RDD computations. Actions triggers execution using lineage graph to load the data into original RDD, carry out all intermediate transformations and return final results to main program or write it out to file system.

III. METHODOLOGY

A. Input Dataset

In this project, the algorithm is implemented for calculation of minimum, maximum and average values of temperature, pressure, humidity and wind speed at a particular location for each day. Data is collected from NCDC web site from years 1996 to years 2016 [3]. 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.

Seven datasets are made from monthly data each comprising of 512MB, 1GB, 2GB, 4GB, 8GB 16GB and 32GB respectively. The data has two formats one for data before 2007 which can be called as old format and other one is data after 2007 which can be called as new format. The difference in format is the order of parameters in a file and some number of other parameters which were added after 2007. But target parameter viz. temperature, pressure, humidity and wind speed are present in both the format.

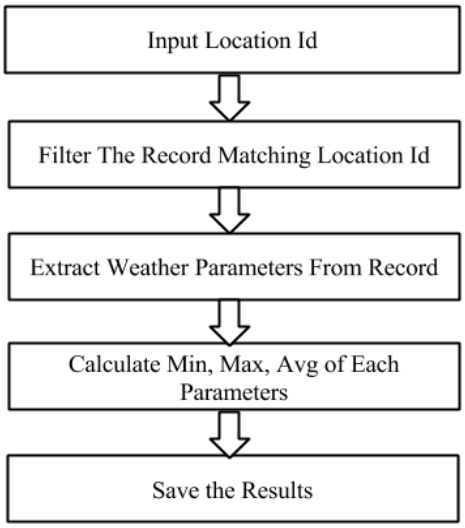

Figure 3 - General Algorithm Flowchart for Weather Data Analysis

B. General Algorithm Flowchart

The general algorithm flowchart is shown in figure 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.

This algorithm is implemented in two popular Big Data technology viz. MapReduce and Spark.

MapReduce Implementation –

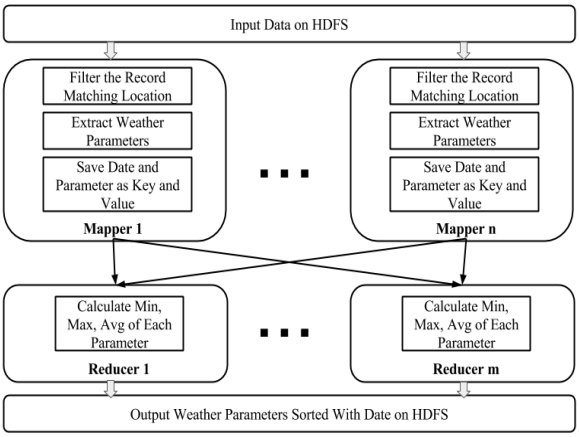

Figure 4 - Algorithm Flowchart for Weather Data Analysis Using MapReduce Implementation

A general algorithm flowchart for weather data analysis using MapReduce is shown in figure 4. A Mapper is run for each block to allow simultaneous processing of each block. Each Mapper filters the record matching the particular location id which is input to analysis module. From the filtered in record, date and all weather parameters are extracted and saved back to the HDFS with date as key and weather parameters as value.

In Mapper phase, memory is allocated only once in setup method for key and value and for each filtered in record, the same memory is reused for each subsequent key and value. This results in the optimization for allocation of memory for each key and value. The combiner is used after each Mapper so that the local minimum, maximum and average values of each weather parameter is calculated. This results in reduction of network traffic and a load on the Reducer.

In the Reducer phase, the global minimum, maximum and average values of each weather parameter i.e. temperature, pressure, humidity and wind speed is calculated. The data is stored back to the HDFS in sorted format with respect to date.

Spark Implementation -

Spark is a fast and general engine for large-scale data processing. It has lot of advantages compared to Hadoop like speed, ease of use, generality and many more. The

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

implementation of weather data analytics using Spark is same as MapReduce implementation explained in above section. It calculates minimum, maximum and average values of weather parameter viz. temperature, pressure, humidity and wind speed.

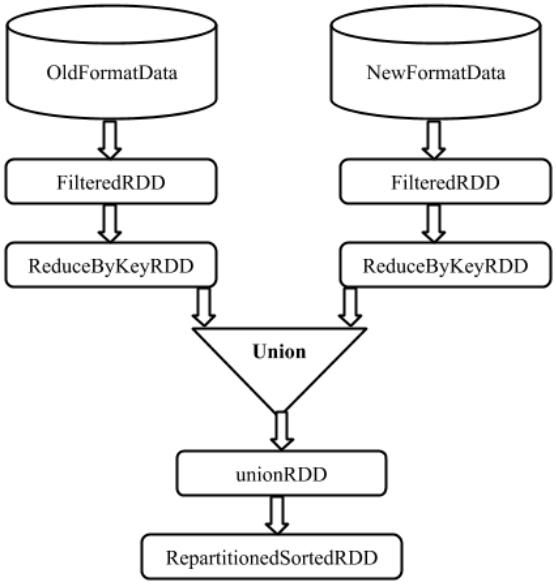

Figure 5 - Algorithm Flowchart for Weather Data Analysis Using Spark Implementation

The general algorithm flowchart for weather data analytics using Spark is shown in figure 5. 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.

IV. RESULTS

Hadoop Cluster Setup -
Hadoop cluster was set up on the three nodes with one namenode and two datanodes. Each node had 16 Intel(R) Xeon(R) CPU E5-2670 (2.6GHz) cores, 64GB RAM and 250GB storage capacity. All the nodes were interconnected by a 1 gigabit Ethernet switch. Hadoop version 2.7.0 was used to setup the cluster.

Benchmarking Results -
The benchmarking was done in three phases. First the comparison between MapReduce implementation using combiner and without combiner on 1 node and then on 2 nodes. This benchmarking was done on two nodes also with assumption that at least some data will flow between network to sense the effect of the combiner. Second, the scalability of Spark code on 2 node for optimizing number of executors. Third, the comparison between MapReduce implementation with Spark implementation on 1 node and 2 node. The result of both the implementation was verified with the actual data given in the dataset for each weather parameter.

Case 1 - MapReduce Code With and Without Combiner –

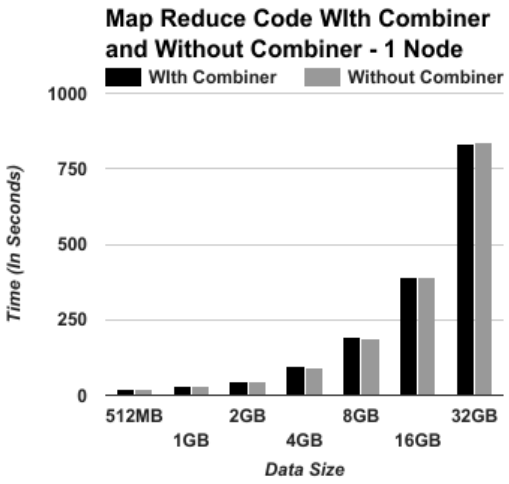

Figure 6 - MapReduce Code Benchmarking for Combiner and Without Combiner on 1 Node

In this benchmarking, MapReduce code was conditionally executed first using combiner and second using without combiner. The implementation has an optional flag for using the combiner or omitting the combiner during the execution. Both execution were executed three times and a time is noted. Then average time is calculated. This is repeated for each dataset from 52MB to 32GB. The graph was plot using each dataset vs average time for with combiner and without combiner. The bar charts are shown in figure 6 and figure 7.

From the chart it can be inferred that on both the benchmarking, the MapReduce implementation with combiner is at least as good as without combiner or slightly better than it. This is more evident in dataset size of 16GB and more on two nodes as shown in figure 7. So for larger dataset there will be more performance benefit using combiner. Hence, MapReduce implementation with combiner is the right approach for weather data analytics.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

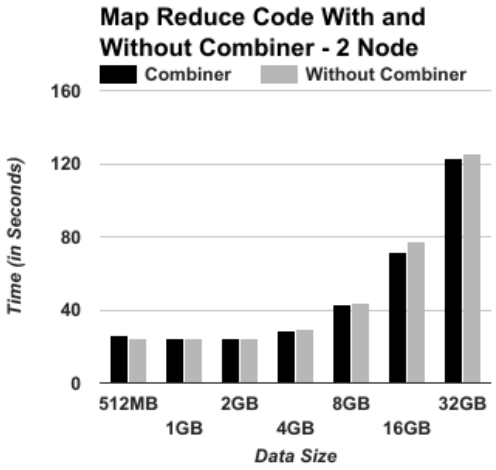

Figure 7 - MapReduce Code Benchmarking for Combiner and Without Combiner on 2 Node

Case 2 - Spark Code with Number of Executors -

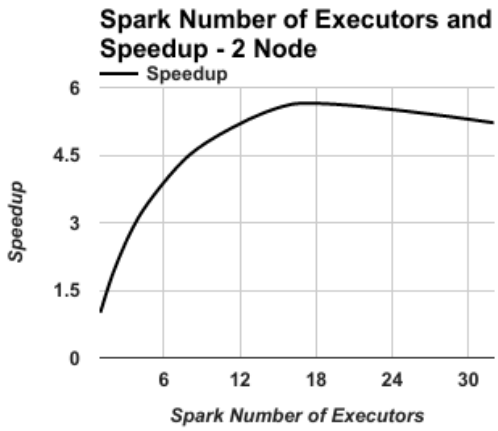

Figure 8 - Spark Benchmarking for Number of Executor vs Speedup

When Spark code is run on a cluster, in this case YARN cluster, the number of executor is passed as argument. The executors are processes of worker node which are in charge of running an individual task of Spark job. Executors are run for the entire lifetime of a Spark application and launched at the start of Spark application. By varying the number of executor on the cluster against the time required, the optimum number of executor can be calculated. The benchmarking is shown in figure 8.

From the graph it can be seen that the optimum number was 14 executors per node. This number of executors were used for running Spark job for further benchmarking. As each node has 16 cores, increasing number of executors more than that is not efficient. Also on each node two cores can be reserved for housekeeping activities.

Case 3 - Spark Implementation with MapReduce Implementation

The Spark code and MapReduce code is run on first 1 Node and then on 2 Node Hadoop Cluster.

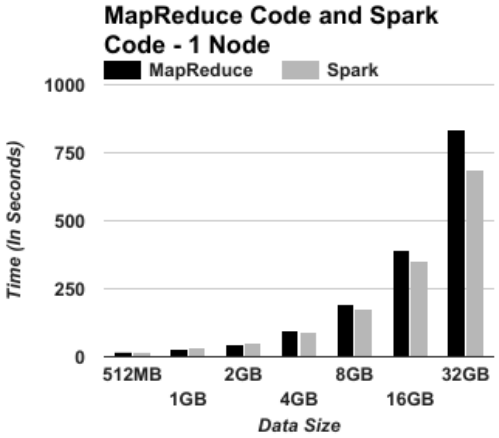

Figure 9 - MapReduce vs Spark Benchmarking on 1 Node

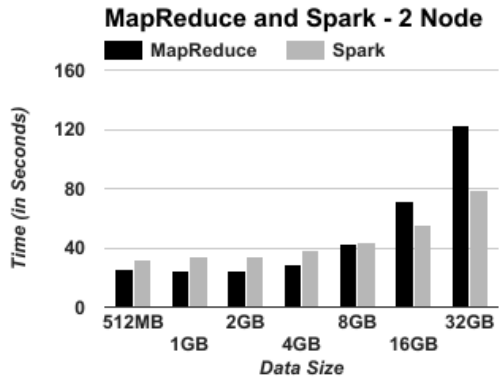

Figure 10 - MapReduce vs Spark Benchmarking on 2 Node

Both Spark and MapReduce code were run on each dataset i.e. 512MB to 32GB. The MapReduce implementation is used with combiner optimization. And the Spark implementation is run with 14 executors per code.
The benchmarking of Spark vs MapReduce code for 1 Node is shown in figure 9 while on 2 Node is shown in figure 10 respectively. From both the chart it can be seen that the performance of Spark is much better than the corresponding MapReduce implementation. Also the performance is more revealed in case when data sizes is more especially 16GB onwards. On 2 node and for 32GB Spark performance is 1.5 times compared to the corresponding MapReduce implementation.

V. RELATED WORK

Riyaz and Sulekha [9] worked on analysisof temperature as weather parameter. They worked on prediction of temperature of a particular city for a particular year. They implemented MapReduce program for the analysis. The execution phase details and results were explained. They also explained the industrial importance of weather data analysis.

P Chouksey at el. [12] has explained the importance of weather data to a human life. It explains the weather parameters which are mostly used for analysis. It suggest

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

that mostly weather data is analyzed using MapReduce. So a comparison study of weather data using MapReduce and Spark is needed.

Shraddha et al. [13] proposed a method for identification of the occurrence of rare patterns in weather using Adaptive technique in data mining. Four major steps of data mining like data collection, data preprocessing, data cleaning, and data transformation and smoothing were explained.

A. Gayathri et al. [14] worked on the study of weather forecasting using data mining techniques. She explained various kind of weather forecasting like Now casting, Short range, Medium range and Long range forecasting.

Veershetty et al. [15] build a Hadoop platform to analyze the weather data with temperature and yearly precipitation as weather parameter. They studied the comparison for weather data analysis using Pig and Hive. They proved that the performance of HIVE is better compared to Pig. They claimed that the analytical engine has capability to scale better in Hadoop cluster.

VI. CONCLUSION AND FUTURE WORK

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 two prominent Big Data technology viz. MapReduce and Spark. The same algorithm is implemented using MapReduce and Spark programming paradigm and their relative performance is studied. The benchmarking of weather data shows that the performance of Spark is very much better than the corresponding MapReduce implementation. In the proposed implementation of Spark has 1.5 x performance than the corresponding MapReduce implementation for weather data analytics. Hence, it can be concluded that the Spark has better performance for weather data analytics than the MapReduce.

ACKNOWLEDGMENT

I would like to acknowledge my Principal at College NRI Institute of Information Science and Technology (NIIST) Bhopal, 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.

REFERENCES

[1] Denissen, Jaap JA, et al. "The effects of weather on daily mood: A multilevel approach" Emotion 8.5 (2008): 662.
[2] Zaslavsky, Arkady, Charith Perera, and Dimitrios Georgakopoulos. "Sensing as a service and big data." arXiv preprint arXiv:1301.0159 (2013).
[3] NCDC Weather Data [Online]. Available: https://www.ncdc.noaa.gov/orders/qclcd/
[4] Katal, Avita, Mohammad Wazid, and R. H. Goudar. "Big data: issues, challenges, tools and good practices." Contemporary Computing (IC3), 2013 Sixth International Conference on. IEEE, 2013.
[5] Apache Hadoop [Online]. Available: http://hadoop.apache.org
[6] Apache Spark [Online]. Available: http://spark.apache.org/
[7] Tudorica, Bogdan George, and Cristian Bucur. "A comparison between several NoSQL databases with comments and notes." 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research. IEEE, 2011.
[8] Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.
[9] Riyaz P.A., Surekha M.V., "Leveraging MapReduce With Hadoop for Weather Data Analytics" IOSR Journal of Computer Engineering, Volume 17, Issue 03, May-June 2015
[10] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." ACM SIGOPS operating systems review. Vol. 37. No. 5. ACM, 2003.
[11] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.
[12] Chouksey P., Chauhan A., "A Review of Weather Data Analytics using Big Data", IJARCCE, ISSN: 2278-1021 Volume-06, Issue-01, Page No (365-368), January, 2017
[13] Miss. Shraddha V. Shingne, Prof. Anil D.Warbhe and Prof. Shyam Dubey, "Weather Forecasting using Adaptive technique in Data Mining", International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 091 - 095
[14] A Gayathri, M. Revathi, J. Velmurugan, "A Survey on Weather forecasting using Data Mining", IJARCCE, ISSN: 2278-1021 Volume-05, Issue-02, Page No (298-300), February, 2016
[15] Veershetty Dagade, Mahesh Lagali,Supriya Avadhani and Priya Kalekar, "Big Data Weather Analytics Using Hadoop",IJETCSE, ISSN: 0976-1353 Volume-14 Issue-02, April 2015

BIOGRAPHIES

Priyanka Chouksey M.Tech Scholar in Computer Science and Engineering at NRI Institute of Information Science and Technology (NIIST), Bhopal, India. She has done B.E. in CSE from Mittal Institute of Technology, Bhopal, India. Her area of research is in Algorithms, Parallel and Distributed Computing, Data Mining and Big Data Analytics.

Abhishek Singh Chauhan An Assistant Professor in Computer Science and Engineering Department at NRI Institute of Information Science & Technology (NIIST), Bhopal, India. He has completed M.Tech in CSE from Samrat Ashok Technological Institute and pursuing PhD from Bhagwant University, Ajmer, India. His main research interests includes Web Application Security & Network Security.