MAPREDUCE ALGORITHM FOR WEATHER DATASET

KHALID ADAM ISMAIL HAMMAD

Master of Science

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : KHALID ADAM ISMAIL HAMMAD

Date of Birth : 19 NOVEMBER 1987

Title : MAPREDUCE ALGORITHM FOR WEATHER DATASET

Academic Session : 2016/2017

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- ☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)	(Supervisor's Signature)
P02179844	Associate Prof. Dr. Mazlina Abdul Majid
New IC/Passport Number	Name of Supervisor
Date:5/5/17	Date:5/5/17

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Computer Science

____

(Supervisor's Signature)

Full Name : DR. MAZLINA ABDUL MAJID

Position : ASSOCIATE PROF

Date : 5/5/2017

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student's Signature)

Full Name : KHALID ADAM ISMAIL HAMMAD

ID Number : MCC14010

Date : 5 May 2017

MAPREDUCE ALGORITHM FOR WEATHER DATASET

KHALID ADAM ISMAIL HAMMAD

Thesis submitted in fulfillment of the requirements
for the award of the degree of Master of in Science

Faculty of Computer Systems & Software Engineering
UNIVERSITI MALAYSIA PAHANG

MAY 2017

ACKNOWLEDGEMENTS

I am grateful and I would like to express my deepest gratitude to God Almighty and whoever supported me to complete this thesis, including my supervisor, attached university, friends, and family. First and foremost, I am thankful to my supervisor Associate Prof. Dr. Mazlina Abdulmajid for her germinal ideas, invaluable guidance, continuous encouragement and unwavering support in making this research possible. She has always impressed me with her outstanding professional conduct, her strong conviction for science, and her belief that MSc. program is only a start of a life-long learning experience. I appreciate her consistent support from the first day I applied to graduate program to these concluding moments. Again, I gratefully acknowledge the immense support and advice from my co-supervisor Prof. Dr. Jasni Mohamad Zain. It has been a great pleasure studying under his supervision and I have learnt a lot from her experience. I am truly grateful for my supervisors' progressive vision about my training in science, their tolerance of my naïve mistakes, and their commitment to my future career.

I would like to thank Dr. Mohammad Adam Ibrahim, who provided me an opportunity to do my research in a very promising field in Big Data. He is an erudite, lenient, frankness and openness to my research points of view and his great ability in grasping the research direction of Big Data. Certainly, he has been able to trigger on me lots of ideas. Definitely, an unusual Dr, hard to find someone like him.

I am greatly indebted to my brother and sister in Islam, Mr Ibrahim Abaker Targio and Mrs Noor Akma Abu Bakar for they assistance, encouragement and the sacrifices they made during this research. I really appreciate for standing by me at all times, may Allah reward them abundantly.

Last but not least, I would like to thank all my friends for their prayers and support. I am also grateful for the insightful comments given by the pre-viva committee: Dr. Norazih Ahamed and Dr. Vitaliy Mezhuyev. I also thank the staff of the Faculty of Computer Systems & Software Engineering for their cooperation throughout this research.

I acknowledge my sincere indebtedness and gratitude to my family for their love, dream and sacrifice throughout my life. I am also grateful to my parent for their sacrifice, patience, and understanding that were inevitable to make this work possible. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals.

Finally, I thank Allah for giving me good health, I am grateful to Allah for being alive to complete this study, I look forward to him to continue to direct me in whatever step I take in life. Praise be to Allah the Lord of the world!!!

ABSTRAK

Ramalan cuaca memainkan peranan yang penting dalam rutin harian manusia, perniagaan dan dalam membuat keputusan. Teknologi dalam bidang ramalan cuaca sedang berkembang dengan pesat kerana keperluannya yang kritikal dalam mendapatkan keputusan ramalan yang tepat. Dari penerokaan literatur, penyelidik telah mendapati bahawa data cuaca adalah penting untuk dianalisis dalam bentuk struktur data. Kebanyakan data kaji cuaca diwakili oleh data tidak berstruktur dengan sifat-sifat yang berbeza seperti suhu, kelembapan, keterlihatan, dan tekanan. Data-data ini diperolehi daripada beberapa jenis sensor. Data cuaca ini bersaiz besar, mempunyai halaju tinggi dan kepelbagaian jenis data yang dapat dilihat dalam ciri-ciri 'Big Data'. Di samping itu, ciri-ciri ini juga menyumbang kepada kerumitan dalam pemprosesan data dan proses ramalan menjadi semakin kompleks. Analisis 'Big Data' merupakan satu konsep baru untuk memproses data yang besar. Konsep baru ini digunakan dalam data cuaca yang akan membantu untuk menyusun semua data kepada data berstruktur. Kaedah yang biasa digunakan dalam menganalisis 'Big Data' adalah model 'MapReduce'. Penggunaan model 'MapReduce' dalam set data pemprosesan cuaca belum diterokai secara meluas. Oleh itu, kajian ini memberi tumpuan kepada analisis dataset cuaca menggunakan algoritma 'MapReduce'. Set data dalam tempoh 10 tahun (1997 hingga 2007) telah digunakan dan ia diperolehi daripada agensi NOAA. Set data ini asalnya disimpan dalam 'Hadoop' sejenis Sistem Fail Teragih. Algoritma 'MapReduce' telah dibangunkan menggunakan pengaturcaraan Java. Algoritma ini telah diuji menggunakan set data yang bersaiz kecil dan besar. Atribut suhu, kelembapan dan keterlihatan telah diekstrak daripada set data oleh algoritma 'MapReduce' ke dalam bentuk struktur data. Analisis bergrafik telah digunakan untuk mewakili hasil daripada algoritma 'MapReduce' ini. Algoritma yang dicadangkan ini telah dibandingkan dengan model sedia ada yang dikenali sebagai model AWK (Alfred Aho, Peter Weinberger, dan Brian Kernighan). Tujuan perbandingan ini adalah untuk menyiasat keupayaan model yang dicadangkan dalam pemprosesan secara selari. Keputusan perbandingan menunjukkan bahawa algoritma 'MapReduce' lebih menjimatkan masa sebanyak 37%, 25% dan 11% kurang berbanding AWK daripada segi masa pemprosesan bagi data bersaiz 10GB, 5GB and 1GB. Hasil kajian ini telah menunjukkan penggunaakan MapReduce algorithm menghasilkan impak yang sangat tinggi dalam ramalan cuaca. Selain itu, hasil daripada MapReduce algorithm juga telah menghasilkan corak yang ketara dalam suhu, kelembapan dan penglihatan dan maklumat ini sangat penting dalam bidang ramalan kajicuaca.

ABSTRACT

Weather forecasting plays a vital role in human daily routine, business and their decisions. The technology for weather forecasting is evolving rapidly due to the critical needs in obtaining the accurate prediction results. From the literature exploration, the researchers have found that weather data is important to be analysed in form of structure data. Most of data in weather is represented in unstructured data with different attributes such as temperature, humidity, visibility, and pressure. These data were captured by different types of sensors. The weather data consists of high volumes, high velocity and variety of data which is reflects to the characteristics of Big Data. In addition, these characteristics also contribute to the complexity on the data processing and prediction. Big Data analytics is a new concept to process the Big Data. For weather data, this new concept will help to organise the data into structure data. The well-known method for Big Data analytics is MapReduce Model. Nevertheless, the usage of MapReduce Model in processing weather dataset is not widely explored. Therefore, this research is focus on analysing the weather dataset using MapReduce Algorithm. The historical dataset in 10 years' period (1997 to 2007) has been used and this dataset is obtained from NOAA. This original dataset is stored in Hadoop Distributed File System. Next, MapReduce Algorithm is developed using Java programming. The algorithm is tested using small and big dataset. The temperature, humidity and visibility attributes from the dataset has been extracted by the MapReduce Algorithm into structure data. Graphical analysis has been used to represent the result from the MapReduce Algorithm. Results from the proposed algorithm have been compared with the existing model known as AWK (Alfred Aho, Peter Weinberger, and Brian Kernighan) model. The purpose of the comparison is to investigate the capability of the proposed model in parallel processing. The comparison results shown that MapReduce Algorithm has produced 37%, 25% and 11% less compared to AWK in term of processing time for 10GB, 5GB and 1GB data, respectively. This result has revealed the significant impact to the used of MapReduce Algorithm in weather prediction. In addition, the MapReduce results have discovered the significant pattern of temperature, humidity and visibility information which is valuable for the weather prediction.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Background

In recent times, people never cease their efforts on predicting the trend of weather changes. Every step forward of weather forecasting technology has great academic and practical significance James et al., (2014). This is not because of changes in climate influences people's daily life, but due to the fact that the advance investigation of weather forecasting reflects and indicate the progress of human's ability to know the earth.

Weather is therefore the most critical for human in many aspects of life. The study and knowledge of how weather evolves over time in some location or country in the world can be beneficial for several purposes. Such knowledge or information could be used for future predictions. For instance, the knowledge of how temperature changes affect the tourists and precipitation aid in flood planning. The terms weather and climate are sometimes used interchangeable in different situations. Their main difference is that weather prediction refers to a short period (e.g. several days to one week), on the other hand, climate prediction involves the process of predicting the future evolution for months, years, etc (Dhanashri, 2015). Major data attributes included in the collected weather from the National Oceanic and Atmospheric Administration (NOAA) information include: year, month, day, temperature, dew point, humidity, significant weather, wind direction, pressure, precipitation snowfall, wind speed, etc.

Many significant research efforts are utilized to develop weather forecasting methods including computational intelligence technologies that have been accepted as appropriate means for weather forecasting and reported encouraging result since 1980s (Chen, 2000 and Kwong et al., 2012). However, the coming of Big Data era brings the opportunities to improve the forecasting accuracy of weather phenomena in advance. Some conventional difficulties in the weather forecasting tasks are expected to be solved with Big Data volume of weather information. Specifically, for weather forecasting tasks, the variation tendency of atmospheric phenomenon is quite unstable and complex, therefore, thousands of related variables are changing every second so that a small change of a certain variable may greatly affect the weather condition.

Unfortunately, the number of variables that can be handled in a certain model is limited. Especially, for computational intelligence models, if too many variables are employed, the overfitting problem is very difficult to be avoided with smaller number of training samples Hong et al., (2008). Accordingly, some fundamental assumptions are required, and the accuracy of the forecasting results highly depends on the correctness of initial condition of the assumptions.

Generally, the conception of “Big Data” refers to the increasing volume of the data set that used to analyze problem in different research domains. Combined with statistical methods and computational intelligence technologies, Big Data has brought a revolution to many traditional research fields including the meteorology, genomics, complex physics simulations, and biological and environmental research, etc. The principles of Big Data are to “let data speaking”, which means, when the volume of data is big enough, the hidden relevance in data set will be revealed via the statistical disciplines. Therefore, if massive weather data is explored, we may avoid using assumptions in our model, and we have the opportunity to directly analyze the correlations hidden in the weather data. Hence, the generalization of the models and accuracy of results are expected to be improved ultimately.

Additionally, Big Data is a term refer to describe the exponential growth for data, both structured and unstructured data, because the data in this context has to do with data that come from many source such as social media, videos, digital pictures, sensors etc. and that make it difficult to use software tools to capture, analysis, manage, and process data within a tolerable elapsed time. Big Data have three characteristics high volume, high velocity and high variety Avita et al., (2013).

According to Bryson "Weather is the original Big Data problem" It has been discussed earlier though any approach is followed; weather forecasting is the initial value problem. Size of initial data increases, accuracy of forecasting increases (Bryson, 2013). Nick Wakeman with reference to Hurricane Sandy stated in his blog about the importance of Big Data in weather forecasting. With the help of available data, three-days out, forecasters predicted within 10 miles where landfall would occur.

According to author it was possible only because of rapidly growing speed and power of computers, and the ability to collect and analyze data faster and more accurately than before, an even bigger disaster was averted (Nick, 2012). According to Nancy Grady the velocity of weather data plays an important role in the development of economy. This weather data can be used by combining it with other disciplines which can generate new opportunities to businesses. Weather, air travel, safety, financial, health, agriculture entrepreneurs are leveraging weather and climate data to build previously impossible business Nancy et al., (2014). The example of a climate corporate is given by the author who sells bad weather insurance to farmers (NASA, 2002).

In addition, the maturity and proliferation of Big Data projects started over quite a few couple of year. Nevertheless, exploiting full potentials of big data is still at a relatively early phase. Emphatically, the term Big Data refers to huge data sets, high volume, high variety and high velocity with structural complexities of managing, storing, analysing and processing. It is increasingly difficult to managing, storing, analysing and processing the data using current conventional techniques Elena et al., (2012). Big data in a short span has generated a whole new industry by supporting architectures with techniques such as MapReduce and Hadoop. Map/Reduce is a programming paradigm which was made popular by Google, where a task is divided and distributed into small portions to a large number of nodes for processing (map), and then the results are summarized into the final answer (reduce). Likewise, Hadoop uses Map/Reduce for data processing.

In this study, Big Data analysis and weather dataset base on MapReduce will be the main focus of research. We present MapReduce algorithm for weather dataset, and that offers not only weather data analysis, but also establish a guideline for researchers on how to analysis Big Data with MapReduce.

1.2. Problem Statement

There is existing widespread belief that Big Data can aid in forecast improvement provided that hidden patterns can be analysed and discovered. According to Richards and King predictions can be improved through data decision-making (Richards & King, 2014). In 2013, Tucker strongly argued that Big Data will soon be predicting our every move (Tucker, 2013), and likewise Einav and Levin in 2013 revealed that Big Data is most commonly sought after for building predictive models in a world of continuous vital statistical forecasting problems.

Therefore, there is a need for the use of a suitable algorithm that can ensure the analysis of the big weather dataset. There have been several efforts made by researchers for weather dataset, however, analysis of the earlier studies in this area of research has revealed some problem/questions. Specifically, the following problems still need to be solved:

1. Weather data is Big Data dataset (unstructured) which requires new technologies to make possible to extract value from it by capturing and analysis process. *How to interpret weather data for forecasting problem?*

2. There is no available algorithm to interpret the Big Data of weather dataset into the format or pattern for the prediction purpose. *Can MapReduce algorithm use for analyzing weather dataset?*

1.3. Research Aim & Objectives

Motivated by a current lack of clear guidance for approaching the field of ‘Big Data with weather’, the main aim of this research is to provide Big Data algorithm for weather dataset. This algorithm has some objectives that give an overview of available weather and Big Data analysis software within the space and to organize this technology by placing it according to the functional components in the Big Data. In order to achieve the aim of this study, some sets of specific objectives have been formulated as follows:

1. To develop MapReduce algorithm for Big Data weather dataset.

2. To validate the MapReduce algorithm using weather dataset.

1.4. Research Scope

This study focuses on designing algorithm for weather dataset in order to make it suitable for forecasting. However, the main scope will be algorithm that are used for the weather dataset to extract (temperature, humidity, visibility and pressure). The present study would only be implemented using the data obtained from National Oceanic and Atmospheric Administration (NOAA) public datasets.

1.5. Significance of The Research

Increasing evidence of climate change worldwide is prompting governments and scientists to take action to protect people and property from its effects. But to take effective action, there is need to know and understand more detail about the weather. Weather Forecasting with Big Data is an imperative to note that the availability of Big Data alone does not constitute the end of problems (Bacon, 2013). A good example is the existence of a vast amount of data on earthquakes, but there is lack of a reliable model that can accurately predict earthquakes (González, 2013). Some existing challenges are related to hypothesis, testing and models utilized for Big Data forecasting Rose et. al, (2013) and identifies as an added concern, the lack of theory to complement Big Data.

In meteorology, scientists rely on the collection and intensive analysis of information to study weather. These methods enable recognition and prediction of weather patterns in order to provide forecasts for people. But today, the rapid expansion of sensors in the network led to the fast increase in weather data growth (Levin & Einav, 2014). Thus, the significant of this study dramatically illustrates Big Data phenomenon and its impact on weather forecasting. Moreover, the traditional techniques are not effective with Big Data (Hassani & Silva, 2015).

1.6. Thesis Contribution to Knowledge

This research has contributed to knowledge in MapReduce algorithm for weather dataset. The specific contributions of this research are:

1. This research proposed MapReduce algorithm for weather dataset, which is found to be effective in weather with Big Data. The algorithm proposed is tested and comparison is made with some algorithm that well reported for modelling of data for predictive purposes. Findings show that, the proposed MapReduce algorithm is efficient and can be used for big weather dataset.

2. The approaches proposed in this research, has shown how weather dataset using the techniques of Big Data analysis. The proposed algorithm has been able to extract big weather historical dataset. Additionally, the proposed MapReduce algorithm in this study was proposed to deal with big weather dataset irrespective of the size of dataset involved. Hadoop includes MapReduce, a distributed data processing model that runs on large clusters of machines. Hadoop MapReduce job mainly has two functions map function and reduce function. The weather forecasting it benefit in any aspect of our life such as decision making.

1.7. Thesis Organization

This thesis is divided into five chapters. The first chapter is an opening chapter that discussed about the background of this study and it has a number of sections that highlight what the thesis proposed to achieve and how it achieved them. The rest of the chapters as contained in the thesis are organized as follows:

Chapter 2: In this chapter, some of the earlier studies reported in the literature are reviewed in order to know what the researchers have proposed earlier with a view to identifying the missing gaps that needed to be filled. Several techniques used for the weather dataset Hadoop/MapReduce and Big Data analysis are discuss in detail in this chapter.

Chapter 3: The approach proposed MapReduce algorithm for weather dataset in this study was discussed and illustrated here. The implementation of the algorithm proposed for the weather dataset. The chart for the proposed approach and the pseudocode are also illustrated and discussed.

Chapter 4: The results are presented in this chapter. The discussion of the results is also analyzed and in order to validate the proposed algorithm, comparisons are made with the results of using other well reported methods based on a number of metrics.

Chapter 5: This thesis is concluded in this chapter. This chapter also contains the summarized steps taken on the algorithm and the limitation of the proposed algorithm. Also discussed in this chapter is what the present study has contributed to knowledge. Some recommendations for further studies are also listed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the studies reported in the literature that relates to the present studies are reviewed. The techniques that are mostly used for Big Data analysis are extensively discussed. The MapReduce algorithm is illustrated in this chapter and subsequently used in the next chapter for weather forecasting with Big Data. Some related work reported in the literature on the weather forecasting with Big Data is reviewed. This enables the gaps that needed to be filled to be identified. This chapter also discusses some challenges of one technique over the others. The java eclipse language is used for the implementation of the MapReduce algorithm that is proposed in this study and HDFS used for the management cluster is also discussed.

2.2 The Original of Big Data

Understanding the main comcept of Big Data and how it evolved into its current stage is very important. Considering the evolution and complexity of Big Data systems, most of the earlier studies are based on a one-sided viewpoint, such as chronology Borkar et al., (2012) or milepost technologies. The original of Big Data is presented in terms of the data size of interest. Big Data is tied tightly to the capability of efficiently storing and managing larger and larger datasets, with size limitations expanding by orders of magnitude. Specially, for each capability improvement, new database technologies were developed, thus, the history of Big Data can be roughly divided into the following stages.

Megabyte to Gigabyte in the 1970s and 1980s, historical business data introduced the earliest ``big data" challenges in moving from megabyte to gigabyte sizes. The urgent need at that time was to house that data and run relational queries for business analyses and reporting. Research efforts were made to give birth to the ``database machine" that featured integrated hardware and software to solve problems. The underlying philosophy was that such integration would provide better performance at lower cost. After a period of time, it became clear that hardware-specialized database machines could not keep pace with the progress of general-purpose computers. Thus, the descendant database systems are software systems that impose few constraints on hardware and can run on general-purpose computers Hu et al., (2014).

Gigabyte to Terabyte in the late 1980s, the popularization of digital technology caused the volumes of data to be expanded into several gigabytes or even to a terabyte, which is beyond the storage and/or processing capabilities of a single large computer system. Data parallelization was proposed to extend storage capabilities and tasks, such as building indexes and evaluating queries, into disparate hardware. Based on this idea, several types of parallel databases were built, including shared memory databases, shared-disk databases, and shared nothing databases, all are induced by the underlying hardware architecture. However, the three types of databases includes, the shared-nothing architecture, built on a networked cluster of individual machines - each with its own processor, memory and disk (Dewitt & Gray 1992) have witnessed great success. Even in the past few years, we have witnessed the blooming of commercialized products of this type, such as (Teradata & Dayton, 2014), Netezza (Netezza & Marlborough, 2013), Aster Data (Aster & Beijing, 2013), Greenplum (Greenplum & San, 2013), and Vertica (Vertica, 2013). These systems exploit a relational data model and declarative relational query languages, and they pioneered the use of divide-and conquer parallelism to partition data for storage.

Terabyte to Petabyte: During the late 1990 when the database community was admiring its ``finished" work on the parallel database, the rapid development of Web 1.0 led the whole world into the Internet era, along with massive semi-structured or unstructured webpages holding terabytes or petabytes (PBs) of data. The resulting need for search companies was to index and query the mushrooming content of the web. Although, parallel databases handle structured data well, they provide little support for unstructured data. Additionally, systems capabilities were limited to less than several terabytes. To address the challenge of web-scale data managemen0.t and analysis, Google created Google File System (GFS) (Ghemawat et al., 2003) and MapReduce (Dean & Ghemawat, 2008) programming model. GFS and MapReduce enable automatic data parallelization and the distribution of large-scale computation applications to large clusters of commodity servers. System running GFS and MapReduce can scale up and out therefore, it can be able to process unlimited data. In the mid-2000s, user-generated content, various sensors, and other ubiquitous data sources produced an overwhelming flow of mixed-structure data, which called for a paradigm shift in computing architecture and large-scale data processing mechanisms. NoSQL databases, which are scheme-free, fast, highly scalable, and reliable, began to emerge to handle these data. In Jan. 2007, Jim Gary, a database software pioneer, called the shift ``fourth paradigm" Hey et al., (2009). He also argued that the only way to cope with this paradigm is to develop a new generation of computing tools to manage, visualize and analyze the data deluge.

Petabyte to Exabyte under current development trends, data stored and analyzed by big companies will undoubtedly reach the PB to exabyte magnitude. However, current technology still handles terabyte to PB data; there has been no revolutionary technology developed to cope with larger datasets. In Jun. 2011, Richard Egan and Roger Marino Corporation (EMC) published a report entitled ``Extracting Value from Chaos".

2.3 Big Data Definitions

"Big Data refers to data volumes in range of exabytes (1018) and beyond" in Kaisler et al., (2013). As define by the Wikipedia, Big Data is an accumulation of datasets usually huge and complex data that it becomes hard to process using database management tools or traditional data processing applications, where the challenges include capturing, storaging, searching, sharing, transfering, analysing, and visualization Mandal et al., (2017).

Sam Madden from Massachusetts Institute of Technology (MIT) wrote "Big Data means too big, too fast, or too hard for existing tools to process" (Madden, 2012). He also explained, the term 'too big' as the amount of data which might be at petabyte-scale data that come from various sources, 'too fast' as the data growth, which is fast and must be processed quickly, and 'too hard' as the difficulties of Big Data that does not fit neatly into an existing processing tool (Madden, 2012).

From PC Mag (popular magazine based on latest technology news), "Big Data refers to the massive amounts of data that collects over time that are difficult to analyze and handle using common database management tools" Ekbia et al., (2015). John Weathington has defined Big Data as a competitive key parameter in different dimensions such as customers, suppliers, new entrants and substitutes. According to him, Big Data creates products which are valuable and unique, and prelude other products from satisfying the same need. He also described, "Big Data is traditionally characterized as a rushing river: large amounts of data flowing at a rapid pace" (Weathington, 2012) and (Graham et al., 2008).

Philip Hunter state that, "Big Data embodies an ambition to extract value from data, particularly for sales, marketing, and customer relations" (Hunter, 2013). Svetlana Sicular has defined Big Data as "high-volume, -velocity and –variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making" (Sicular, 2013). Big Data refers to datasets that are both big and high in variety and velocity, which makes them difficult to handle using traditional tools and techniques Janssen et al., (2017).

Big Data is a popular term used to describe the exponential growth, availability and use of information, both structured and unstructured Peter et al., (2014). The term "Big Data" is often used to describe massive, complex, and real-time streaming data that requires sophisticated management, analytical, and processing techniques to extract insights (Gupta & George, 2016). Table 2.1 shows summarize the Big Data definitions from view by any author.

Table 2.1 In this table, we summarize the Big Data definitions from view by any author

Definations	**Challenging for traditional application/ Requires new froms of application**	**Big volume of data**	**Competitive key parameter**	**Enhanced insights generator**	**High-volume. High-velocity. High-variety**	**Volum, velocity, variety, veracity**	**Heterogeneous, Autonomous, Complex and Evolving (HACE) Therom**
Kaisler, et al., (2013)		√					
Mandal et al., (2017).	√						
(Madden, 2012)	√				√		
Ekbia et al., (2015)	√						
(Weathington, 2012)			√		√		
Doctorw, 2008) and Weathington, 2012)	√						
(Hunter, 2013)				√			
(Sicular, 2013)				√	√		
(IBM, 2012)						√	
(Wu, et al., 2013)							√
Janssen et al., (2017)	√				√		
Peter et al., (2014)		√					
(Gupta & George, 2016)	√	√					

The table above summarizes some of the Big Data definitions, most of the authors have agreed that Big Data are difficult to process using conventional analysis tools. Additionally, the authors admitted that Big Data have high volume, high velocity and high variety as characteristics.

2.4 Big Data Characteristics

The characteristics of Big Data are well defined by Gartner (Beyer & Laney 2012). The three Vs (volume, velocity and variety) are known as the main characteristics of big data. The characteristics are described in Figure 2.1.

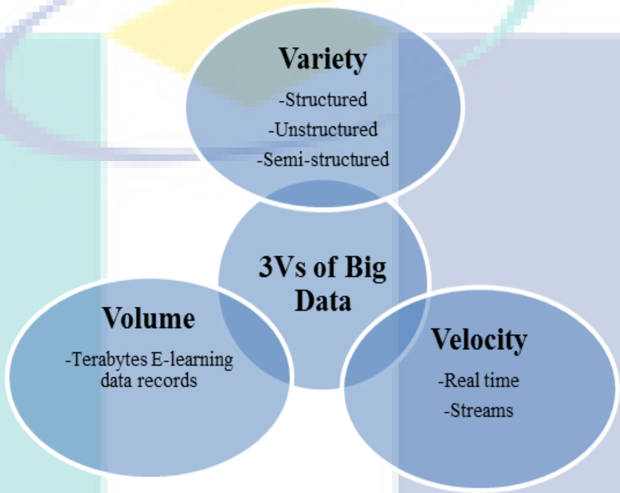


Figure 2.1. 3Vs of Big Data

2.4.1 Volume

Refers to amount of data, and there are many factors that can be contributed to the increase of volume of data such could be as a result of hundreds of terabytes or even petabytes of information generated from everywhere Avita et al., (2013). The number of sources of data for an organization is growing. More data sources consisting big datasets increase the volume of data, which needs to be analyzed Kaisler et al., (2013). Figure 2.1 shows that the data volume is growing from megabytes (106) to petabytes (1015) and beyond. Figure 2.2 indicates that the volume of data stored in the world would be more than 40 zettabytes (1021) by 2020.

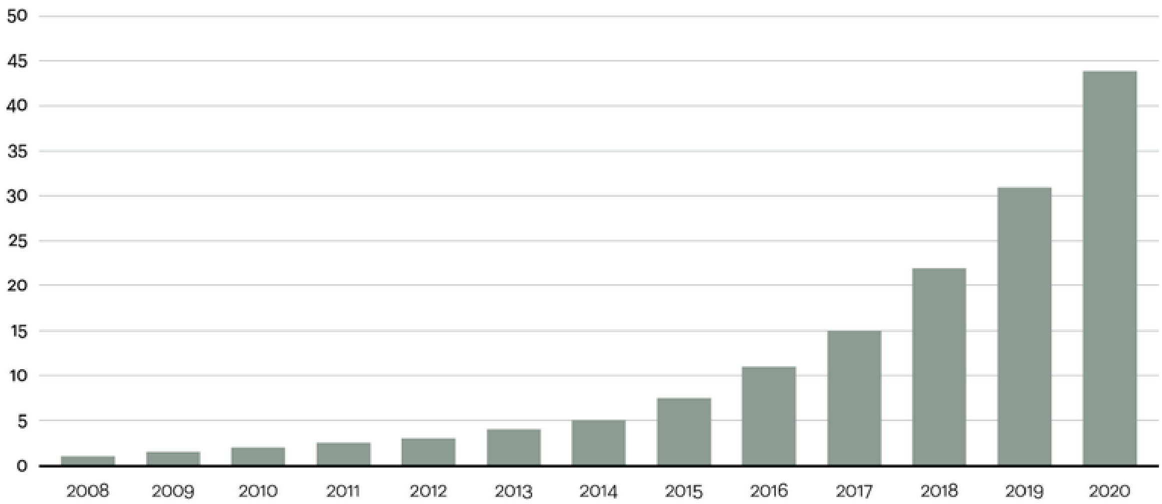


Figure 2.2. Data volume growth by year in zettabytes

Source: Matsuoka et al., (2014).

2.4.2 Velocity

Refers to data speed that measures the velocity of information creation, gushing and collection Kaisler et al., (2013). According to Svetlana Sicular from Gartner, velocity is the most misunderstood Big Data characteristic (Sicular, 2013). She describes that the data velocity is also about the rate changes, and about combining data sets that are coming with different speeds. The velocity of data also describes bursts of activities, rather than the usual steady tempo where velocity frequently equated to only real-time analytics (Sicular, 2013).

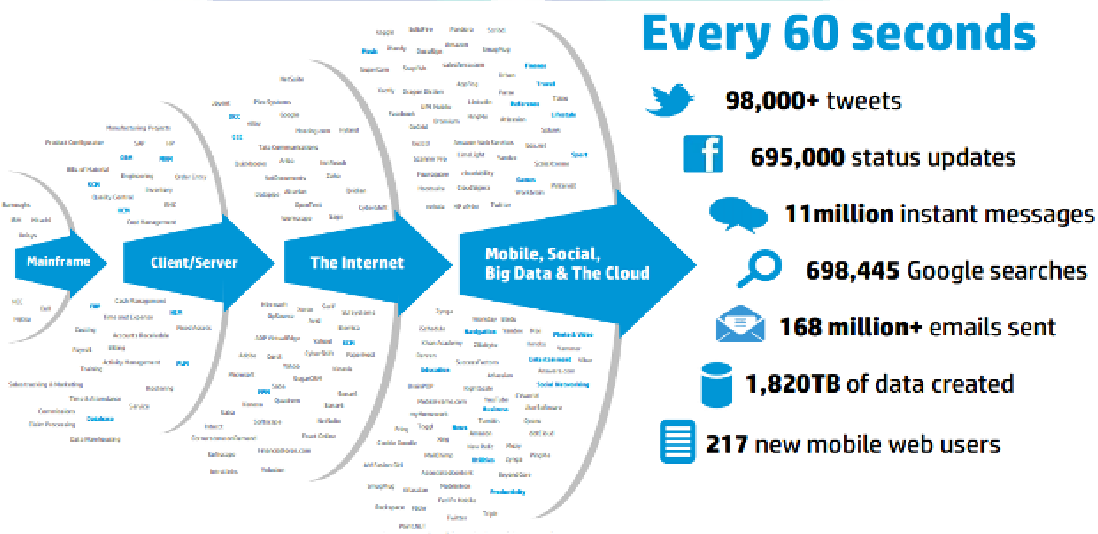


Figure 2.3. Examples of Big Data velocity

Source: Sinanc (2013).

Figure 2.3 shows few examples of the pace of data. Data speed administration is significantly more than a bandwidth issue; it is additionally an ingest issue Kaisler et al., (2013). Figure 2.1 also reflects velocity as a characteristic of Big Data, showing how it requires near real-time and/or real-time analytics.

2.4.3 Variety

Apart from typical structured data, Big Data contains text, audio, images, videos, and many other unstructured and semi-structured data, which are available in many analog and digital formats. From an analytics perspective, variety of data is the biggest challenge to effectively use it. Some researchers believe that, taming the data variety and volatility is the key of Big Data analytics (Infosys, 2013). Figure 2.4 shows the comparison between increment of unstructured, semi-structured data and structured data by years. Figure 2.2 also reflects the increment in verity of data.

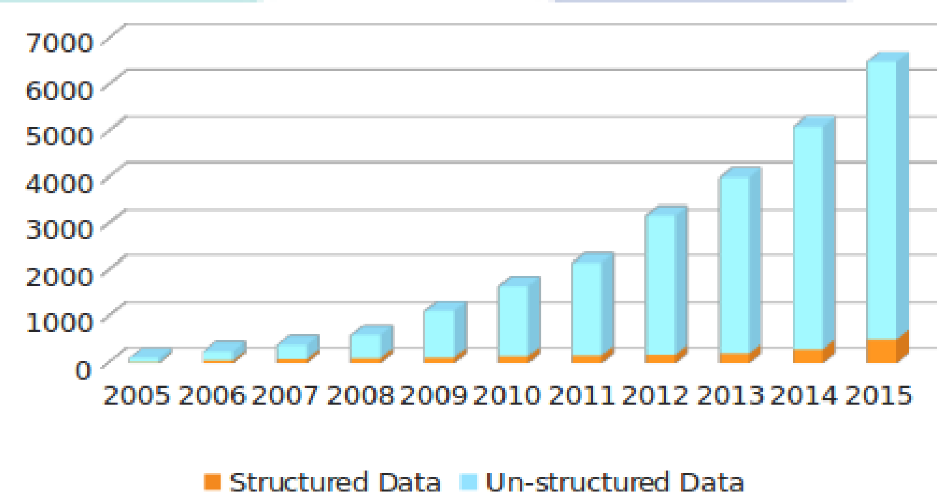


Figure 2.4. Growth of data variety by years

One of the Big Data vendors, IBM has coined additional V for the Big Data characteristics, which is veracity. By veracity, they address the inherent trustworthiness of the data. As Big Data will be used such as for decision making, it is important to make sure that the data can be trusted. Some researchers mentioned ‘viability’ and ‘value’ as the fourth and the fifth characteristics leaving ‘veracity’ out (Biehn, 2013).

2.5 Big Data Analysis

In 2012 the world produced about 2.5 Exabytes of data daily (Mcafee & Brynjolfsson, 2012). Recent study estimates that 7 Zettabytes of data to be generated in 2014 Villars et al., (2011). This data is generated due to the explosion in the use of electronic devices such as computers, sensor networks, and smart phones, as well as the use of social communication sites in several daily activities. This huge amount of generated data needs to be handled using novel and efficient data management systems giving rise to the term "Big Data." This term is currently used to represent such huge and complex data sets that no traditional data processing systems can handle efficiently. Big Data, according to McKinsey Manyika et al., (2011), refers to datasets whose sizes are beyond the abilities of typical database software tools to capture, store, manage and analyze. Big Data needs new technologies that will be able to extract value from those datasets; such processed data might be used in other fields such as artificial intelligence, data mining, etc.

Big Data concern large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences. As humans explore the real world through scientific research, humans unravel the mysteries in the information world through Big Data, which are attracting much attention in academia.

In February 2011, "Science" published "Dealing with Data" album, and jointed Science: Signaling, Science: Translational Medicine and Science: Careers to launch related topics, that discuss the importance of data for scientific research. In May, McKinsey published "Big Data: the next frontier for innovation, competition, and productivity", analyzed application potential of Big Data in different industries from the economic and commercial dimensions, spelled out the development policy for the Government and industry decision makers dealing with Big Data (Olson & Riordan, 2012).

The research and development of Big Data involve national security, life and health, climate variation, geological survey, disaster prevention and reduction and Smart

Planet, which are all associated with spatial data.For example, in the United States, government established National Information Infrastructure in 1993, released National Broadband Plan in 2010, and invested 200 million dollars to start Big Data Research and Development Initiative. In this plan, US Geological Survey and National Aeronautics & Space Administration are most closely associated to spatial data Kim et al., (2014).

The USGS John Wesley Powell Center for Analysis and Synthesis provides scientists with the place and time to indepth analysis, the most advanced computing ability and the collaboration tools to perceive Big Data sets, which eventually promotes the innovation thinking of the Earth System Science (ESS). In this center, scientists cooperate to synthesize the comprehensive and long-term data, and further convert Big Data and the big ideas of Earth science theory to scientific discoveries with the aim to improve the understanding of the ESS and response capabilities. For example, species respond to climate change, earthquake recurrence rate and the next generation of ecological indicators.

2.6 Case Study Weather Forecasting

Weather changes are one of the major problem in the world. For example, Malaysia faced cool temperature in January 2014 resulted several diseases occurs, such as flu. In Thailand, due to this change, more than three peoples die due to cool temperature less than 10 Celsius. It is not a new thing because in 1993 George et al., (1993). Based on upper quote, it could be concluded; the weather changes involved changes in climate condition in a series of data .The detection of this event could involve three major threshold – velocity (how rapid the changes in weather data changes), variety (it is possible the changes in a series of data affected other data) and veracity (the effect of velocity and variety could change the nature of the data).

Weather Climate change definition clearly stated in New England Aquarium (NEAQ) as “Climate describes the average or typical conditions of temperature, relative humidity, cloudiness, precipitation, wind speed and direction, and other meteorological factors that prevail globally or regionally for extended periods. Weather describes the hourly or daily conditions that people experience each day. Which is why it’s often said that “Climate is what you expect; weather is what you get.” .

In practical terms, even a small improvement in forecasting quality can produce enormous benefits for individuals, businesses, and society, from providing warnings for short-term events such as tornados and floods to long-term issues such as how to construct buildings and design infrastructure. For instance, before Hurricane Sandy slammed the northeastern U.S. in October 2012, the European Center for Medium range Weather Forecasting (ECMWF) had successfully predicted the storm track and intensity of the event five days out, while National Weather Service (NWS) models lagged by about a day. The deviation in modelling focused attention on the perceived deficiencies of the NWS.

As a crucial part of this study, climate data should be defined as precise as it could. There are several open source data sets that could be used (public domain). NOAA (https://www.ncdc.noaa.gov/)- NOAA is responsible for preserving, monitoring, assessing, and providing public access to the Nation's treasure of climate and historical weather data and information .This data comes in text dataset, visual and processed, e.g. a mean temperature for year to year. This source has been chosen due to :

1. Availability of data for all places in USA.
2. Availability of data for more than 10 years.
3. The data represent in text, with no visualization and complex presentation.

2.7 Related Work on Weather Forecasting

A number of studies have been reported on efforts made by researchers at improving the weather temperature prediction system. When we talk about weather forecasting with Big Data, numerous research focus on this matters (Halim, Baig, & Bashir, 2006; MacAlpine et al., 2010). these research is still in progress since they found a hole between weather forecasting and Big Data analysis.

Kyger acknowledges that the episode was a "disappointment" for the National Weather Service (NWS). This led, in May 2013, to the U.S. Congress approving $23.7 million in supplemental funding to upgrade NWS systems from 90 teraflops to upward of 200 teraflops, as well as addressing other issues. However, U.S. forecasting technology continues to generate concerns. "There have been a number of important forecasts where U.S. prediction systems performed in an inferior way," Mass says. A recent blog posted by Mass stated that the U.S. had slipped into fourth place in global weather prediction, behind facilities in continental Europe, the U.K., and Canada (Samuel, 2014).

Y.Radhika and M.Shashi presents an application of Support Vector Machines (SVMs) for weather prediction. Time series data of daily maximum temperature at location is studied to predict the maximum temperature of the next day at that location based on the daily maximum temperatures for a span of previous n days referred to as order of the input. Performance of the system is observed for various spans of 2 to 10 days by using optimal values of the kernel (Radhika & Shashi 2013).

Earlier study analyzed the relation between the tourists' stay in Austria and the weather from 1060 to 2012. The number of sunny days in summer and the temperature had a positive effect on the domestic tourists' stay, and precipitation had a negative effect. The foreign tourists' stay was positively affected by sunshine and temperature. This study used the average weather information and tourist demand information in July and August, and the spatial scope of the analysis was the country. However, this study targeted the daily weather forecast in the areas surrounding the beach and very limited areas (Falk, 2014).

The second previous (Kulendran & Wong, 2005) study performed the seasonality analysis of the tourist demand of UK and the Greek tourists in Austria using the quarterly data. The tourist demand was categorized into holiday tourists, business tourists, and total tourists, and two ARIMA models were used to create the prediction model. The study performed the seasonality analysis and forecast using the quarterly data, but it did not offer summer or seasonal insights. Also, its spatial scope was the country. However, its analysis was limited to the daily data in small areas and presented the weather and floating population relation in summer. Table 2.2 shows the existing Big Data analytics examples and their impact on value creation.

Table 2.2 In this table, we summarize the existing Big Data analytics examples and their impact on value creation

Description (Source)	Big Data Analytics Case	Impact on Value Creation
Analyzing patient, characteristics with combination results of medications Manyika et al., (2011).	Analyzing Big Data datasets in a suitable time.	Reduce, under-treatment and over-treatment.
Analyzing, physician entries, and compare them with guidelines, to caution for potential blunders Manyika et al., (2011).	Analyzing, different data Sources, including Unstructured, data sources such as X-ray images.	Reduce adverse, reactions and lower treatment, rates and liability, claims.
State-of-the-art cross selling Manyika et al., (2011).	Analyzing, different (exclusive) data, sources including multi-structured, data sources.	Easier access, to additional Customer, information eventually, leading to an increase, in sales.
Sensor-driven, operations in manufacturing, Manyika et al., (2011).	Realtime, analyzing granular, data originating from, sensors (IoT)	Improved transaction, efficiency, due to ubiquitous process control, and factory optimization.
Customer, ad targeting, based on behavior (Schmarzo, 2011).	Sets of users, based upon behavior, demographics, etc. is formed.	Targeted, sales improve, the transaction, efficiency.
Automated crime intelligence (Genovese & Prentice, 2012).	Analyzing different, (exclusive) data, sources Including, multi-structured, data sources	More-accurate, intelligence and reliable, solution.
Listening, to thousands of customers, (Marr, 2015).	Analyzing, different large, datasets with feedback, originating from web, email and text messages	Automated, tagging of feedback, saves field managers, work hence, time.

Table 2.2 continued

Description (Source)	Big Data Analytics Case	Impact on Value Creation
Electricity consumption forecasting (Wang, 2013).	Analyzing big dataset of the electricity consumption.	Developed prediction model is able to provide sound portability and feasibility in terms of processing Big Data relating to electricity.
Fraud, detection on population Bloem et al., (2012).	Analyzing, all bank Transactions, for criminal behavior.	Better, insights in criminal, behavior, and the ability, to prevent it.
Evolution criteria of weather attributes in Jordan over the evaluated period of time and how it is connected to climate change Jararweh et al., (2014).	Analying Large-Scale Climate data from Jordan climate station.	Results showed that humidity and dew point weather attributes are going to face a significant increase in the future.
Energy Forecasting for Event Venues: Big Data and Prediction Accuracy Grolinger et al., (2016).	Analyzing large variations in consumption caused by the hosted events.	The daily data intervals resulted in higher consumption prediction accuracy than hourly or 15-min readings, which can be explained by the inability of the hourly and 15-min models to capture random variations.
A Floating Population Prediction Model in Travel Spots using Weather Big Data Lee, et al., (2015).	Analysis big dataset of the changes in the floating population based on weather factors.	to predict real daily floating populations in July and August, in South Korea.

The table above summarizes the Big Data analysis and the impact value creation. All these cases are used Big Data analytics, along with the pressure to improve performance and decision making. Moreover, the challenges and the benefits expect to realize by deploying Big Data analytics for needs.

2.8 Hadoop Open Source for Big Data Analysis

Hadoop "is the kernel of the delivering so as to cut edge venture information warehousing" cloud-confronting architectures, MPP, in-database investigation, blended workload administration and a cross breed stockpiling layer (Kobielus, 2012). Designed by Doug Cutting, the maker of Apache Lucene, Hadoop gives a complete toolset for the distributed processing of Big Data sets across clusters of computers using simple programming models, including data analysis, data storage and coordination White et al., (2010).

Hadoop produced from Apache Nutch, is an open source software framework for storing data and running application on clusters of commodity hardware. After realizing the traditional enterprise data warehouse cannot handle big volumes of structured and unstructured data more efficiently than Hadoop, Because Hadoop is open source and can run on commodity hardware, the initial cost savings are dramatic and continue to grow as your organizational data grows. Additionally, Hadoop has a strong Apache community behind it that continues to contribute to its advancement Ghemawat et al., (2003), to general society. Almost a year later all Nutch algorithms had been ported to utilize MapReduce and HDFS.

In 2006, Nutch turned into a different subproject under the name Hadoop and after two years it turned into a top-level venture at Apache, confirming its success. During that year, Hadoop was utilized by numerous universal companies, for example, Facebook. Hadoop is a synonym for Big Data because of its capabilities to store and handle huge amounts of (unstructured) data within a smaller timeframe in an economically responsible way (Kuil, 2012). For this reason the Hadoop ecosystems play a major role in Big Data analytics.

With traditional data analytics, only the peak can be analyzed and utilized in order to create value or support value creation. This peak often consists of highly structured data stored in traditional data warehouses. Since the amount of unstructured data is growing rapidly as described earlier, this peak is becoming relatively smaller. With Hadoop, it is possible to store and analyze unstructured data in a much smaller timeframe using the power of distributed and parallel computing on commodity hardware. More essential, the line demonstrating the limit of information that can be used and information that can not be use, is dropping, prompting a much more prominent top and consequently, in more conceivable quality. Figure 2.5 shows New technologies make it possible to utilize more data.

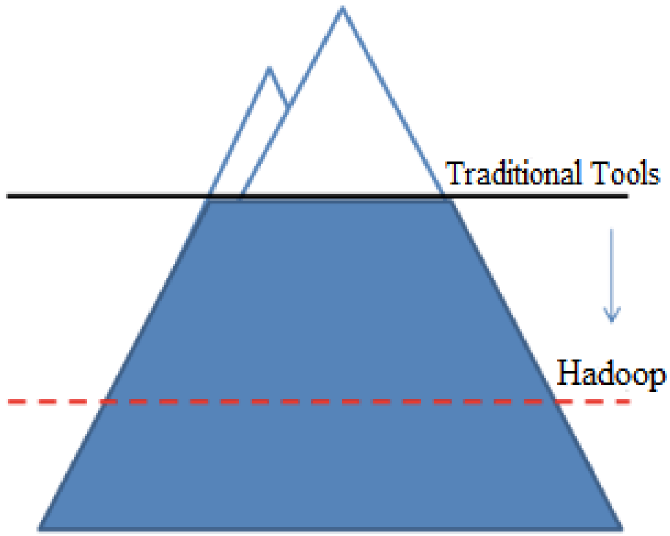


Figure 2.5. New technologies make it possible to utilize more data
Source: Jain et al., (2016).

Together with its free license, huge community and open source systems, many initiatives using Hadoop have been emerged, also indicating its success. Also, many big IT organizations started to distribute their own commercial version of Hadoop by adding enterprise support, additional functionalities and tools and even bundled with specific hardware (which contradicts with the fact that Hadoop is great because it runs on cheap commodity hardware). Examples are Cloudera, EMC GreenPlum, IBM InfoSphere BigInsights, Amazon AWS Elastic MapReduce and Microsoft's release of Hadoop on its cloud platform Azure by porting Hadoop to Windows.

A critical piece of the Hadoop ecosystem is the Hadoop Distributed File System (HDFS) making the partition of data and computing across many nodes as possible. Albeit frequently seen as a NoSQL database, it is definitely not. Albeit both data storage systems use distributed storage across multiple nodes, there are critical contrasts. Firstly, NoSQL databases is responsible for maintaining records by providing efficient ways to insert, modify or delete records using indexed attributes. HDFS does not index data, rather when Hadoop executes a job, it scans all data which is considered very inefficient within the field of NoSQL databases. This does not matter for Hadoop since its performance is mainly depending on the number of nodes and the sum of raw CPU power (Knulst, 2012). Another distinction between NoSQL databases and Hadoop in general usefulness of the database: NoSQL database focusses on running little jobs as quickly as would be prudent (e.g. Giving data to a site being asked for) while Hadoop focuses on running big jobs utilizing big dataset.

2.8.1 Hadoop Distributed File System (HDFS)

Hadoop Distributed File System Shafer et al., (2010) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance typically has a single Namenode, the core of the HDFS file system that keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. DataNodes stores data in the HDFS and a functional filesystem has more than one DataNodes, with data replicated across them. The file system uses the TCP/IP layer for communication, while clients use Remote Procedure Call (RPC) to communicate between each other. Figure 2.6. shows HDFS Architecture.

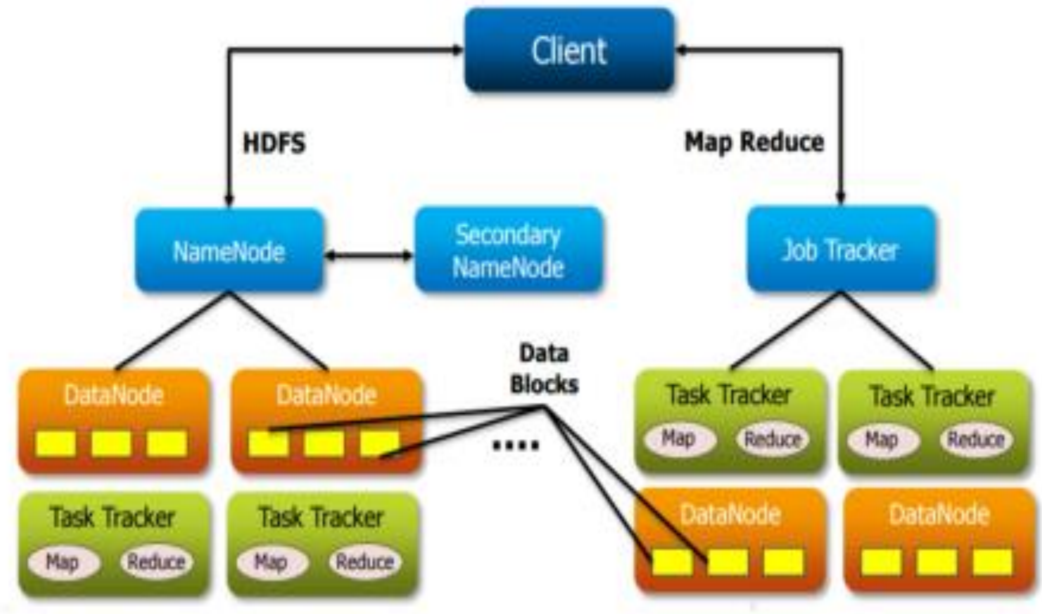


Figure 2.6. Architecture of HDFS

Source: Wang et al., (2016).

HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines. It achieves reliability by replicating the data across multiple hosts. With the default replication value data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can talk to each other to re- balance data and to move copies around. HDFS main feature is the ability to scale to a virtually unlimited storage capacity by simply adding new machines to the cluster at any time. The default distributed file system (HDFS) stores file in blocks of 64 MB. It can store files of varying size from 100 MB to GB. Hadoop architecture contains the Name node, data nodes, secondary name node, task tracker and job tracker. Name node maintained the Metadata information about the block stored in the Hadoop distributed file system. Files are stored in blocks in a distributed manner. The Secondary name node does the work of maintaining the validity of the Name Node and updating the Name Node Information time to time.

Data node actually stores the data. The Job Tracker actually receives the job from the user and split it into parts. Job Tracker then assigns these split jobs to the Task Tracker. Task Tracker runs on the Data node they fetch the data from the data node and execute the task. They continuously talk to the Job Tracker. Job Tracker coordinates the job submitted by the user. Task Tracker has fixed number of the slots for running the tasks. The Job tracker selects the Task Tracker which has the free available slots. It is useful to choose the Task Tracker on the same rack where the data is stored this is known as rack awareness. With this inter rack bandwidth can be saved. Figure 2.7 shows the arrangement of the different component of Hadoop on a single node. In this arrangement all the component Name Node, Secondary Name Node, Data Node, Job Tracker, and Task Tracker are on the same system. The User submits its job in the form of MapReduce task. The data Node and the Task Tracker are on the same system so that the best speed for the read and write can be achieved.

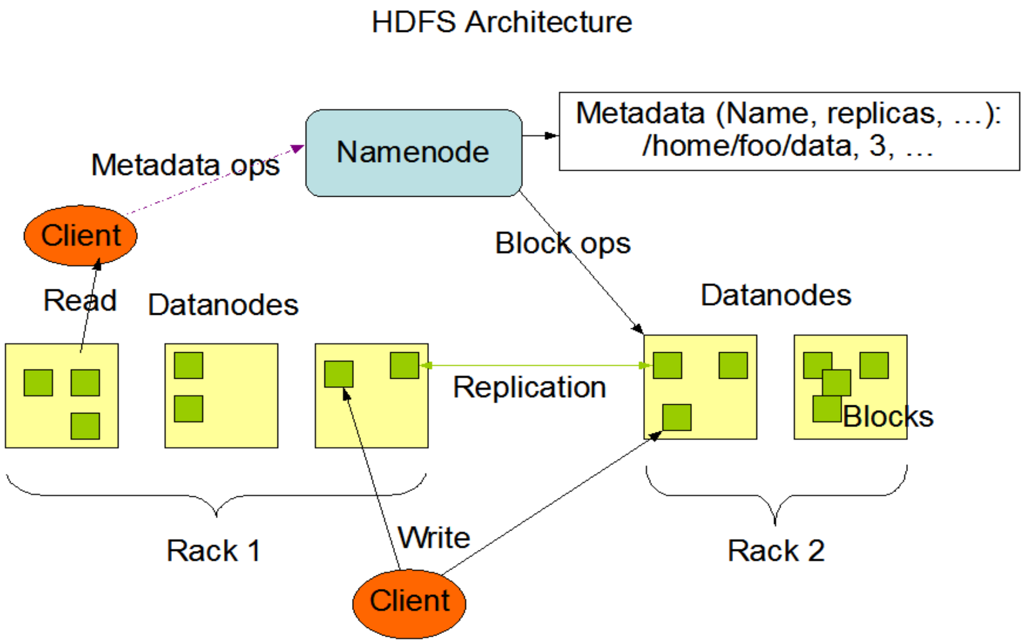


Figure 2.7. Hadoop on a single node
Source: Kornacker et al., (2015).

2.8.1.1 Storage in HDFS Filesystem

HDFS is a distributed file system with master/slave architecture and built in Hadoop platform. It is designed for storing large files which could be hundreds of megabytes, gigabytes and petabytes in size; and running on clusters of computers which could be inexpensive and not necessary to be highly reliable commodity hardware Barbosa et al., (2015). In a cluster, the HDFS are consisted of a single Namenode (the master) and a cluster of DataNodes (the slaves). DataNodes store the data of the filesystem and retrieve blocks when the Namenode tells them to. They report their status to the Namenode periodically. There is also a secondary Namenode which produce snapshot of the primary NameNode's memory structures to avoid the problems brought by file system corruption. The HDFS clusters are setup at the beginning of the process and then transfer the collected data sets from the local system to HDFS for the future sentiment analysis. Figure 2.8 shows process and how to store datasets into HDFS and Figure 2.9 shows the process and how to querying from cluster.

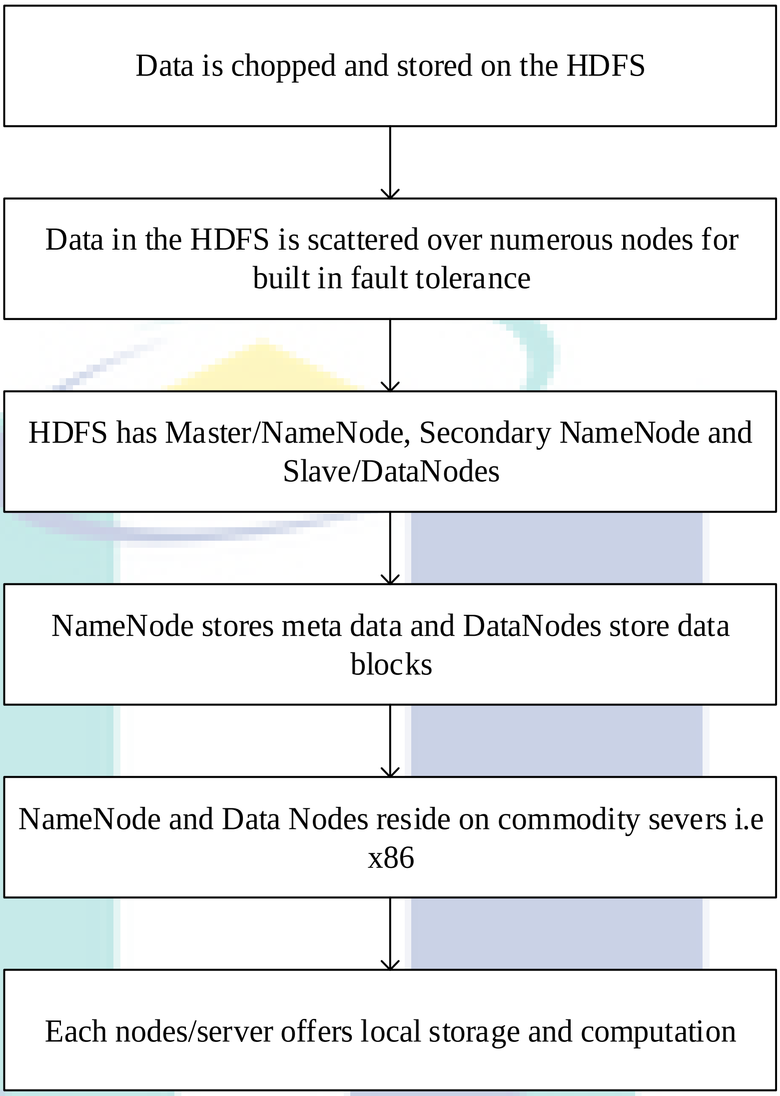


Figure 2.8. Data storage in Hadoop

Source: Ujjal Marjit et al., (2015).

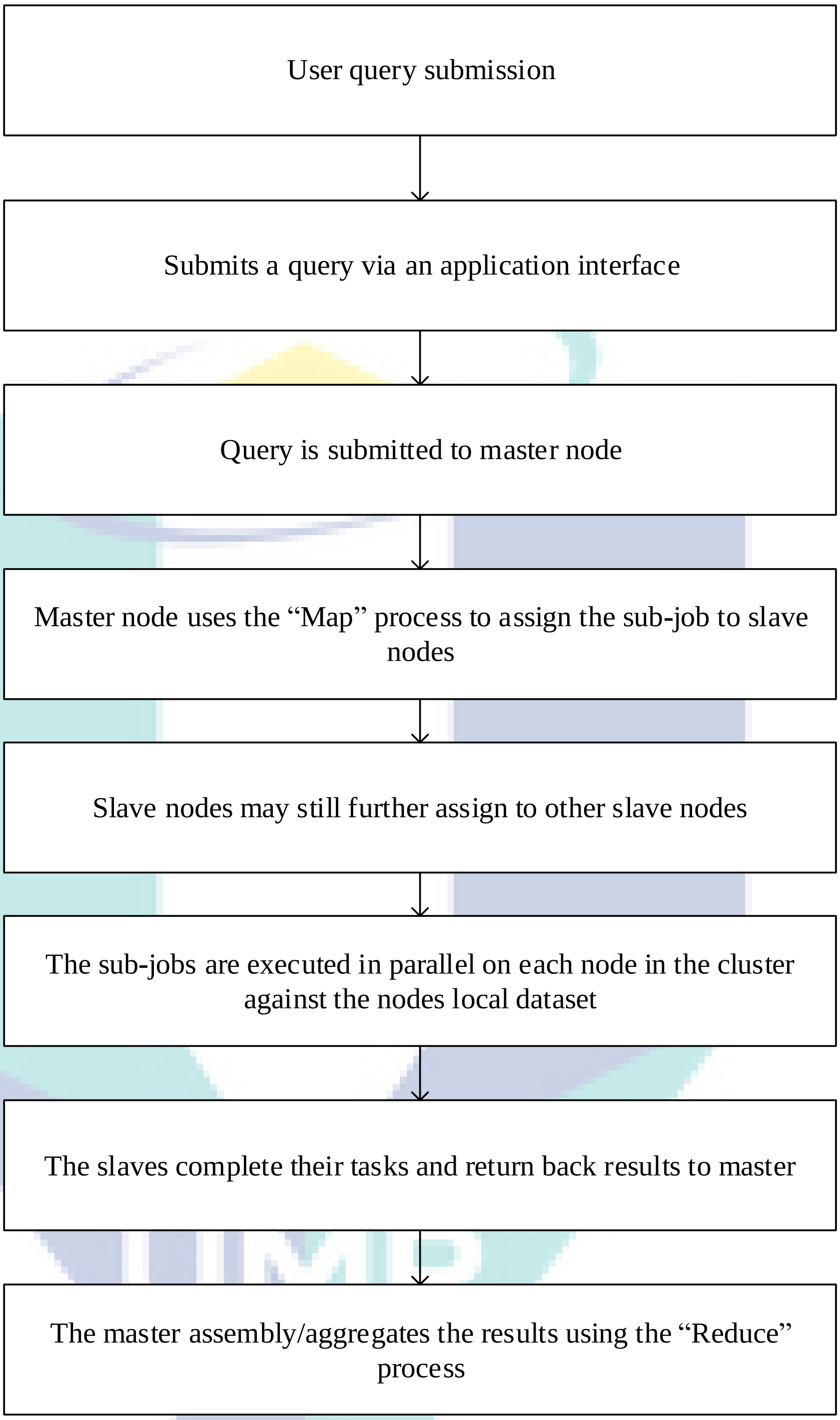


Figure 2.9. Querying Big Data HDFS

Source: Marjit et al., (2015).

2.8.2 MapReduce

The MapReduce programming model simplifies the complexity of running parallel data processing functions across multiple computing nodes in a cluster, by allowing a programmer with no specific knowledge of parallel programming to create MapReduce functions running in parallel on the cluster Jiang et al., (2015). MapReduce automatically handles the gathering of results across the multiple nodes and returns a single result or set. More importantly, the MapReduce runtime system offers fault tolerance that is entirely transparent to programmers. Figure 2.10 shows the overall process of MapReduce.

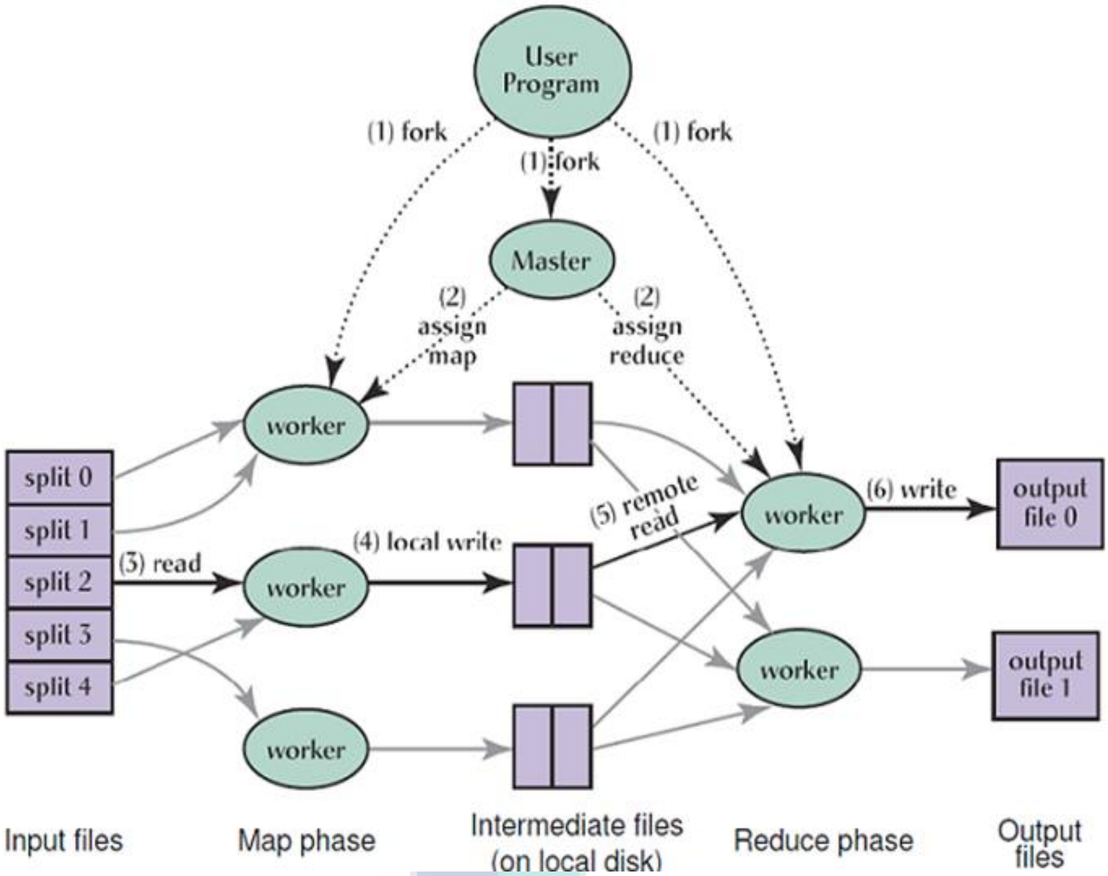


Figure 2.10. The overall process of MapReduce application
Source: Devine (2011).

The term MapReduce refers to two separate and distinct tasks. First is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples (Dean & Ghemawat, 2004). As the sequence of the name MapReduce implies, the reduce job is always performed after the map job. Putting the Map and Reduce functions to work efficiently requires an algorithm too. The standard steps for a MapReduce work-flow shows in Figure 2.11.

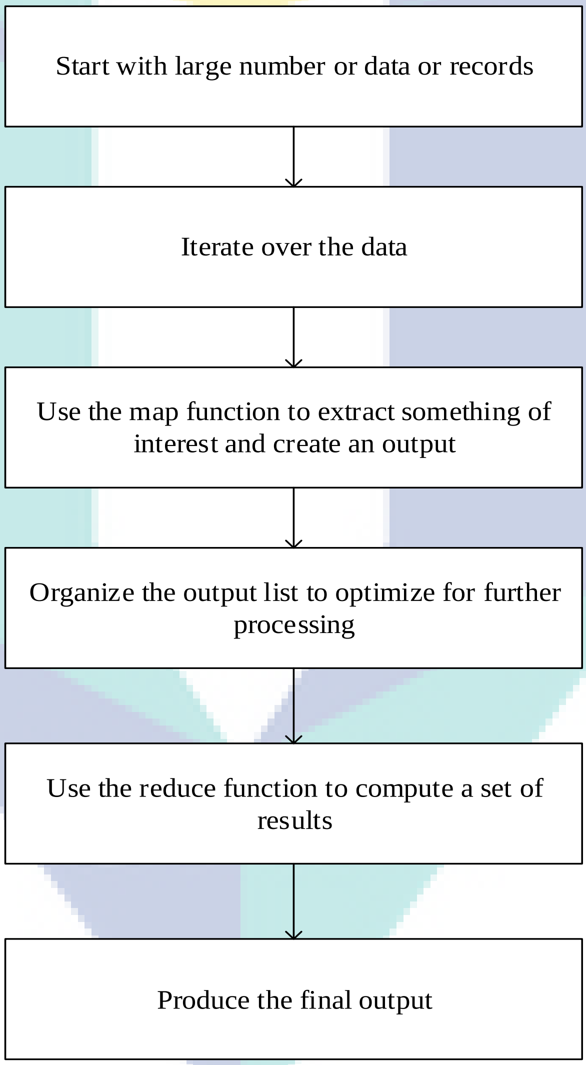


Figure 2.11. MapReduce steps

Source: Tang et al., (2015).

2.8.2.1 MapReduce Model

Each MapReduce application has two major types of operations - a map operation and a reduce operation. MapReduce allows for parallel processing of the map and reduction operations in each application. Each mapping operation is independent of the others, meaning all mappers can be performed in parallel on multiple machines. In practice, the number of concurrent map operations is limited by the data source and/or the number of CPUs near that data. Similarly, a set of reduce operations can be performed in parallel during the reduction phase. All outputs of map operations that share the same key are presented to the same reduce operation Francisci et al., (2010). Although the above process seemingly inefficient compared to sequential algorithms, MapReduce can be applied to process significantly larger datasets than "commodity" servers. For example, a large computing cluster can use MapReduce to sort a petabyte of data in only a few hours. Parallelism also offers some possibility of recovering from partial failure of computing nodes or storage units during the operation (Yoon & Kim, 2011). In other words, if one mapper or reducer fails, the work can be rescheduled, assuming the input data is still available. Input data sets are, in most cases, available even in presence of storage unit failures, because each data set normally has three replicas stored in three individual storage unites. Figure 2.12 shows eight datanodes replication data.

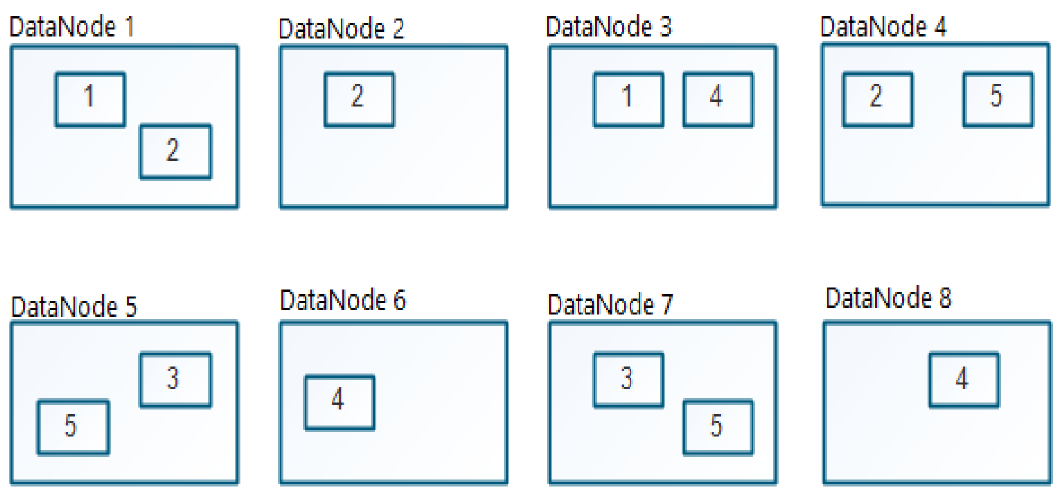


Figure 2.12. Hadoop data replication on data datanodes

Source: Robert et al., (2010) .

2.8.2.2 MapReduce Execution Process Steps

MapReduce Algorithm uses the following three main steps:

- Map Function
- Shuffle Function
- Reduce Function

Discuss here is the role and responsibility of each function in MapReduce algorithm. A simple word counting example used to explain them in-detail.

A. Map Function

Map Function is the first step in MapReduce Algorithm. It takes input tasks (Datasets) and divides them into smaller sub-tasks. Then perform required computation on each sub-task in parallel. This step performs the following two sub-steps:

- Splitting

  Splitting step takes input Dataset from Source and divides into smaller Sub-Datasets.
- Mapping

  Mapping step takes those smaller Sub-Datasets and performs required action or computation on each Sub-Dataset. The output of this Map Function is a set of key and value pairs as <Key, Value> as shows in Figure 2.13.

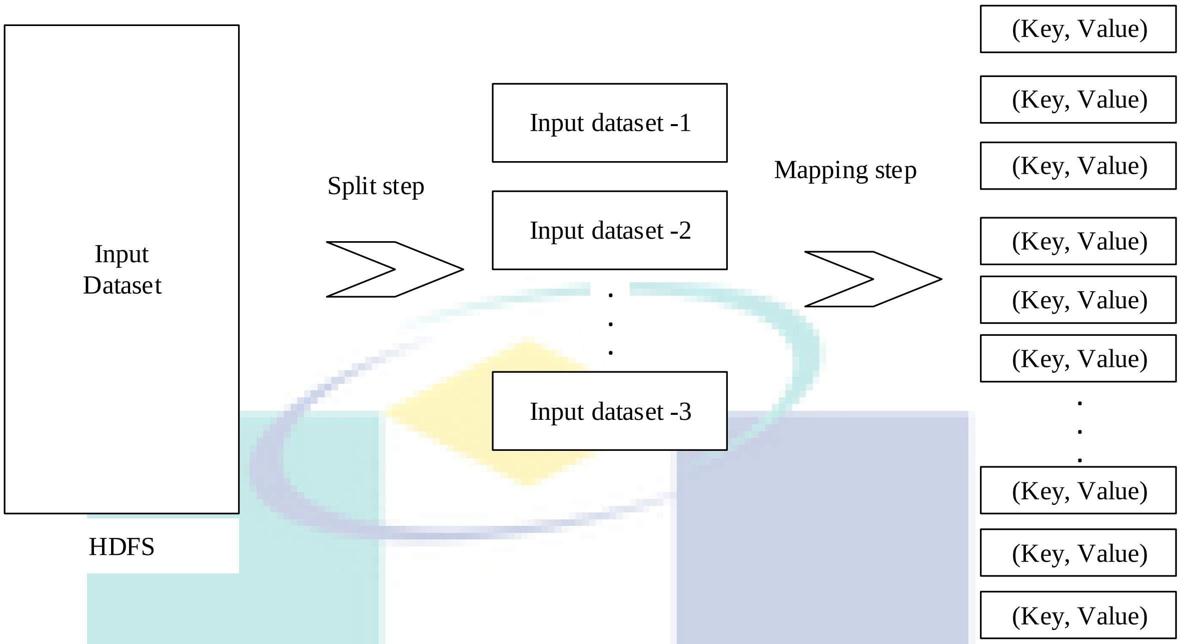


Figure 2.13. Map function

Source: Ghemawat, (2004)

B. Shuffle Function

It is the second step in MapReduce Algorithm. Shuffle Function is also known as “Combine Function” as shows in Figure 2.14. It takes a list of outputs coming from “Map Function” and performs these two sub-steps on each and every key-value pair.

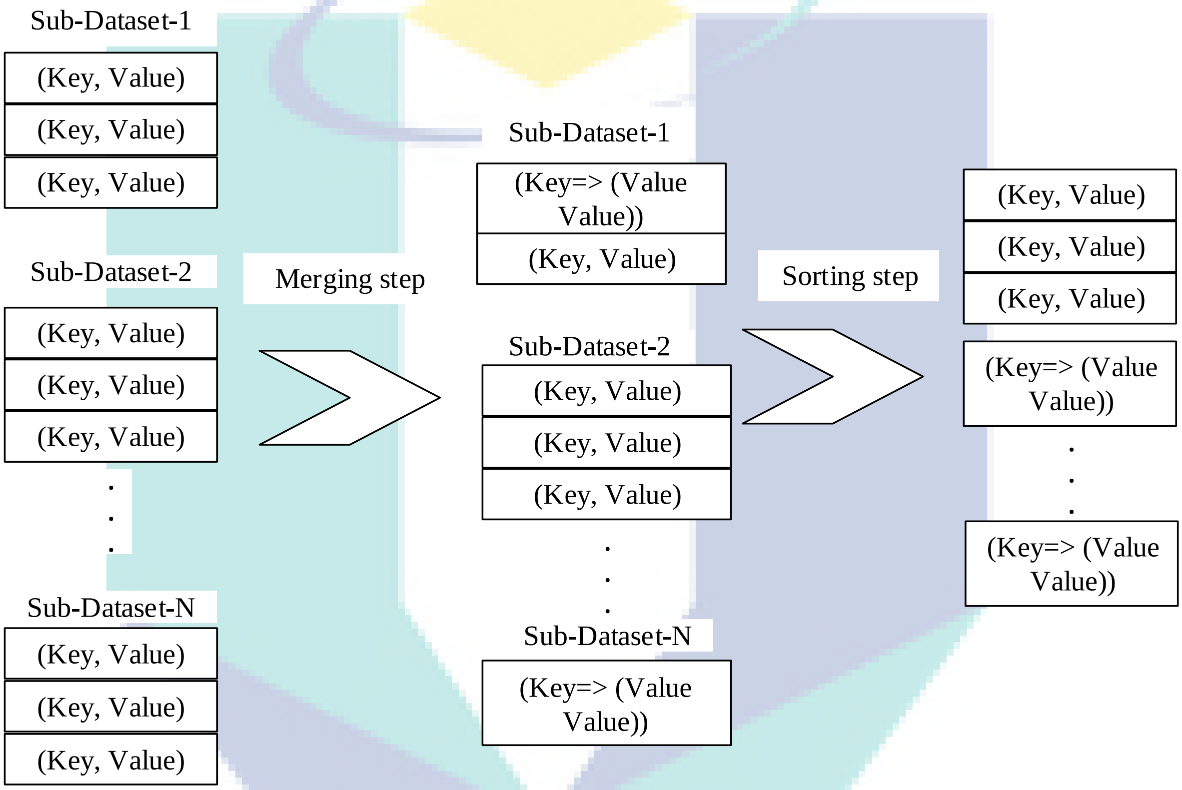


Figure 2.14. Shuffle function

Source: Ghemawat, (2004).

- Merging

  Merging step combines all key-value pairs which have same keys (that is grouping key-value pairs by comparing “Key”). This step returns <Key, List<Value>>.

- Sorting

  Sorting step takes input from Merging step and sort all key-value pairs by using Keys. This step also returns <Key, List<Value>> output but with sorted key-value pairs. Finally, Shuffle Function returns a list of <Key, List<Value>> sorted pairs to next step.

C. Reduce Function

It is the final step in MapReduce Algorithm. It performs only one step: Reduce step. It takes list of <Key, List<Value>> sorted pairs from Shuffle Function and perform reduce operation as show in Figure 2.15.

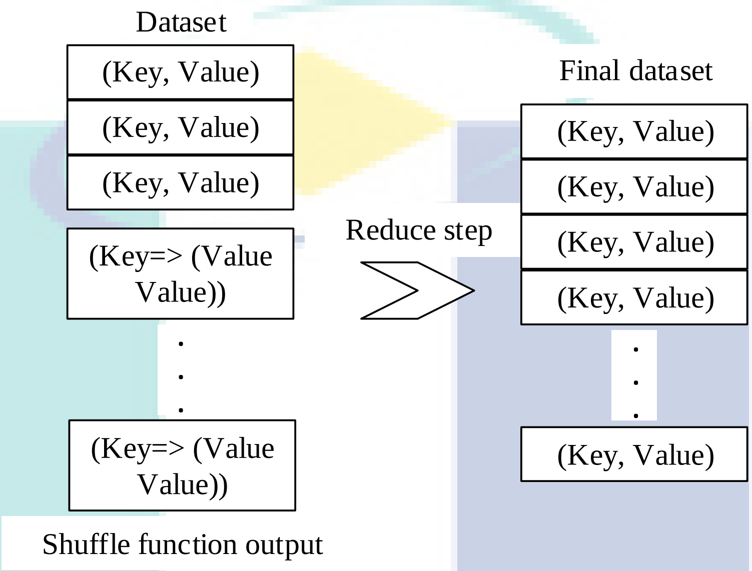


Figure 2.15. Reduce function

Source: Ghemawat, (2004)

2.9 Conclusions

Big Data is defined as a big amount of data which requires new technologies to make possible to extract value from it by capturing and analysis process. Analytics often involves studying past historical data to research potential trends. Weather prediction has been one of the most interesting and fascinating domain and it plays a significant role in meteorology. Weather prediction is to estimate of future weather conditions. Weather condition is the state of atmosphere at a given time in terms of weather variables like rainfall, thunderstorm, cloud conditions, temperature, pressure, wind direction etc. Predicting the weather is essential to help preparing for the best and the worst of the climate.

The knowlede gathered through this literature review show that it possible extract weather Big Data to using for forecasting purpose. The present reseach, extract the the Big Data weather (unstrectured) data for forecasting purpose. Beside that, we discussion the Hadoop/MapReduce. In chapter 3, we discuss the MapReduce algorithm. However, Hadoop is a distributed, scalable, and portable file-system that is use in our use cases with MapReduce.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter provides a methodological approach of the study. The fundamental stages, process and steps of the method for executing the objectives of this research have been addressed in this chapter. The methodology has been represented to show the individual steps and sequential processes used. This chapter also begins with the proposed approach which involves the design algorithm and implementation was used for analyzing the big weather dataset. Also, this chapter discusses the weather dataset that was retrieved in an online open repository. Moreover, this chapter explains the steps and the implementation of the MapReduce algorithm. The details of this chapter has been discussed below.

3.2 The Proposed Approach

The proposed approach focuses on designing the MapReduce algorithm. Figure 3.1 shows the research framework for this study that leads to the final product.

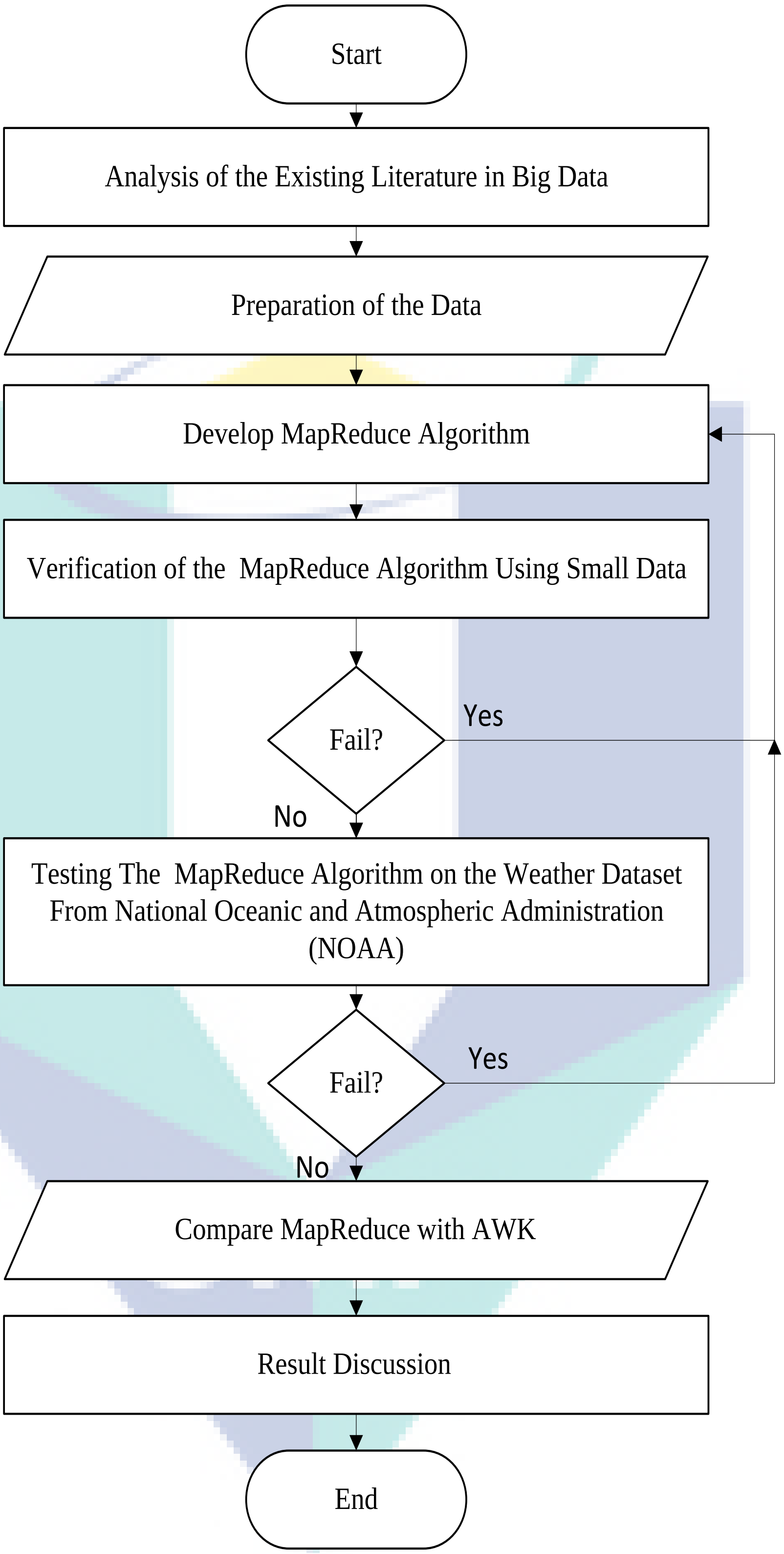


Figure 3.1. Approach for this study that leads to the final product

3.3 Big Data Weather Dataset

The National Oceanic and Atmospheric Administration (NOAA) Data Centers (of which NCDC is the largest) are world-class centers that provide long-term preservation, management, and ready accessibility to environmental data. The combined archive includes records taken even before Ben Franklin's weather observations and continues with the latest real-time satellite imagery. The Centers are part of the National Environmental Satellite, Data and Information Service (NESDIS). The NCDC is located in Asheville, NC. The data that are used was obtained from the National Oceanic and Atmospheric Administration (http://www.noaa.gov/). The data is stored using a line-oriented ASCII format, in which each line is a record. The format supports a rich set of meteorological elements, many of which are optional or with variable data lengths. This study focuses on the elements, such as temperature, Visibility and Humidity which are always present and are of fixed width. Table 3.1 shows a sample line with some of the salient fields highlighted. Data files are organized by date and weather station. There is a directory for each year since 1997 to 2007, each containing a zipped file for each weather station with its readings for that year. Since there are tens of thousands of weather stations, the whole dataset is made up of a big number of relatively small files, more details of dataset in Appendix C.

DATABASE: Worldwide surface observations (hourly/synoptic).
Datasets - TD3280 and TD9956.
TD3280--Navy and first order National Weather Service (NWS) stations.
Data Type- ASCII character data.
Quality Control -Undergoes extensive automated and manual QC.

- About 380 stations currently active.

- Includes most surface elements observed in the U.S. (wind speed and direction, temperature, dew point, cloud data, sea level pressure, altimeter setting, station pressure, present weather, and visibility). Wind gust, daily precipitation amount, and snow depth are not included, but are placed in TD3210. Hourly precipitation amount stored in separate dataset (TD3240).

- "Specials" are not included and only synoptic hours (every 3rd hour) are included for (for most stations).

Table 3.1 Weather Dataset into ASCII Format

Wban Number, YearMonthDay, Time, Station Type, Maintenance Indicator, Sky Conditions, Visibility, Weather Type, Dry Bulb Temp, Dew Point Temp, Wet Bulb Temp, % Relative Humidity, Wind Speed (kt), Wind Direction, Wind Char. Gusts (kt), Val for Wind Char., Station Pressure, Pressure Tendency, Sea Level Pressure, Record Type, Precip. Total

03013,19960701,0053,AO20,-,CLR ,10SM ,-,64,60.1,35, 87 , 7 ,180,-,0 ,26.30,-,162,AA,-

03013,19960701,0153,AO20,-,CLR ,10SM ,-,64.9,60.1,35, 84 , 10 ,190,-,0 ,26.30,6,153,AA,-

03013,19960701,0253,AO20,-,CLR ,10SM ,-,62.1,60.1,34.9, 93 , 8 ,200,-,0 ,26.29,-,150,AA,-

03013,19960701,0353,AO20,-,CLR ,10SM ,-,60.1,59,34.7, 96 , 3 ,310,-,0 ,26.29,-,151,AA,-

03013,19960701,0453,AO20,-,CLR ,10SM ,-,59,57.9,34.6, 96 , 0 ,000,-,0 ,26.30,5,154,AA,-

03013,19960701,0553,AO20,-,CLR ,10SM ,-,64,61,35, 90 , 0 ,000,-,0 ,26.30,-,155,AA,-

03013,19960701,0653,AO20,-,CLR ,10SM ,-,66.9,62.1,35.2, 84 , 6 ,310,-,0 ,26.31,-,162,AA,-

03013,19960701,0753,AO20,-,CLR ,10SM ,-,72,63,35.4, 73 , 5 ,310,-,0 ,26.31,3,160,AA,-

3.4 Algorithm for the Big Weather Dataset

As shown in Figure 3.2, the algorithm comprised of three sections: the input, output and the technique used. The Algorithm consists of two functions: a map function and a reduce function, and when a function is called the steps of actions take place. The first procedure is to pre-process the input weather dataset split into a number of pieces of a specified size. The input data are aggregated from NOAA for the purpose of analysis during the process. The second procedure is mapped by the mapping function (line 1-5 of the Algorithm). In the procedure, the processed data is used to create an exact model (key and value), and then choosing the temperature, visibility and humidity are independent variable. The third procedure is to reduce the key and value (line 6- 12 of Algorithm). In the procedure, the final step in Reduce the (key, value) Residual for the final model. Then previous procedures are repeated by using transformed variable in the final procedure (line 7-12 of Algorithm). If not, the final model is returned.

The mapper emits an intermediate key-value pair for each weather file. The reducer sums up all counts for each temperature.

```
              Input:   D // Dataset
             Output: Temp, Visib, Hum // temperature, Visibility and Humidity
Begin
1: Class M
2:      Method M (LongWritable, T1, T2, IntWritable)
3:      for all T1, IntWritable do
              Emit (string temp; line 10)
              Emit (string Visib; line 8)
              Emit (string Hum; line 13)
4:   If stringUtils is Numeric then
5:   Output (datepart, temp, visib, hum)
End.
Begin
6: class R
7:      Method Reduce (T1, IntWritable, T2, IntWritable)
8:      Sum of all Temps, Visib, Hum key
9:      NumItems
10:     While (values.hasNext()
```

```
Sum Temps, Visib, Hum += values.next().get()
numItems += 1
        Output.collect(key, new IntWritable(sumTemp / numItems)
        Output.collect(key, new IntWritable(sum Visib / numItems)
        Output.collect(key, new IntWritable(sum Hum / numItems)
End.
```

Figure 3.2. Proposed algorithm

3.5 MapReduce Algorithm Stages

In this section, we provide a comprehensive assessment of the MapReduce algorithm and weather dataset. As shown in Figure 3.3, the user gets the weather unstructured dataset from the NOAA, these data will be extract and then loaded into the Hadoop distributed file system. Then the stored dataset will be transferred Map/Reduce Algorithm. The MapReduce have three stages map, shuffle and then reduce. Such mechanism can help reduce the amount of time to process Big Data unstructured weather datasets. The output of the weather dataset gain from MapReduce processing will be combined and use to generate reports. With that being said, in the next subsection, we provide in the detailed description of MapReduce Algorithm.

3.5.1 MapReduce

MapReduce enables an inexperienced programmer to develop parallel programs and create a program that can use computers in a cloud. In most cases, programmers are required to execute only two functions, namely, the map (mapper) and reduce functions (reducer), which are commonly utilized in functional programming. The mapper regards the key/value pair as input and generates intermediate key/value pairs, and the reducer merges all pairs associated with the same (intermediate) key and then generates an output. The map function is applied to each input (key1, value1), in which the input domain is different from the generated output pairs list (key2, value2). The elements of the list (key2, value2) are then grouped by a key. After grouping, the list (key2, value2) is divided into several lists [key2, list (value2)], and the reduce function is applied to each list [key2, list (value2)] for generating a final result list (key3, value3). (Input) <k1, v1> -> Map -> <k2, v2> -> Combine -> <k2, v2> -> Reduce -> <k3, v3> (Output).

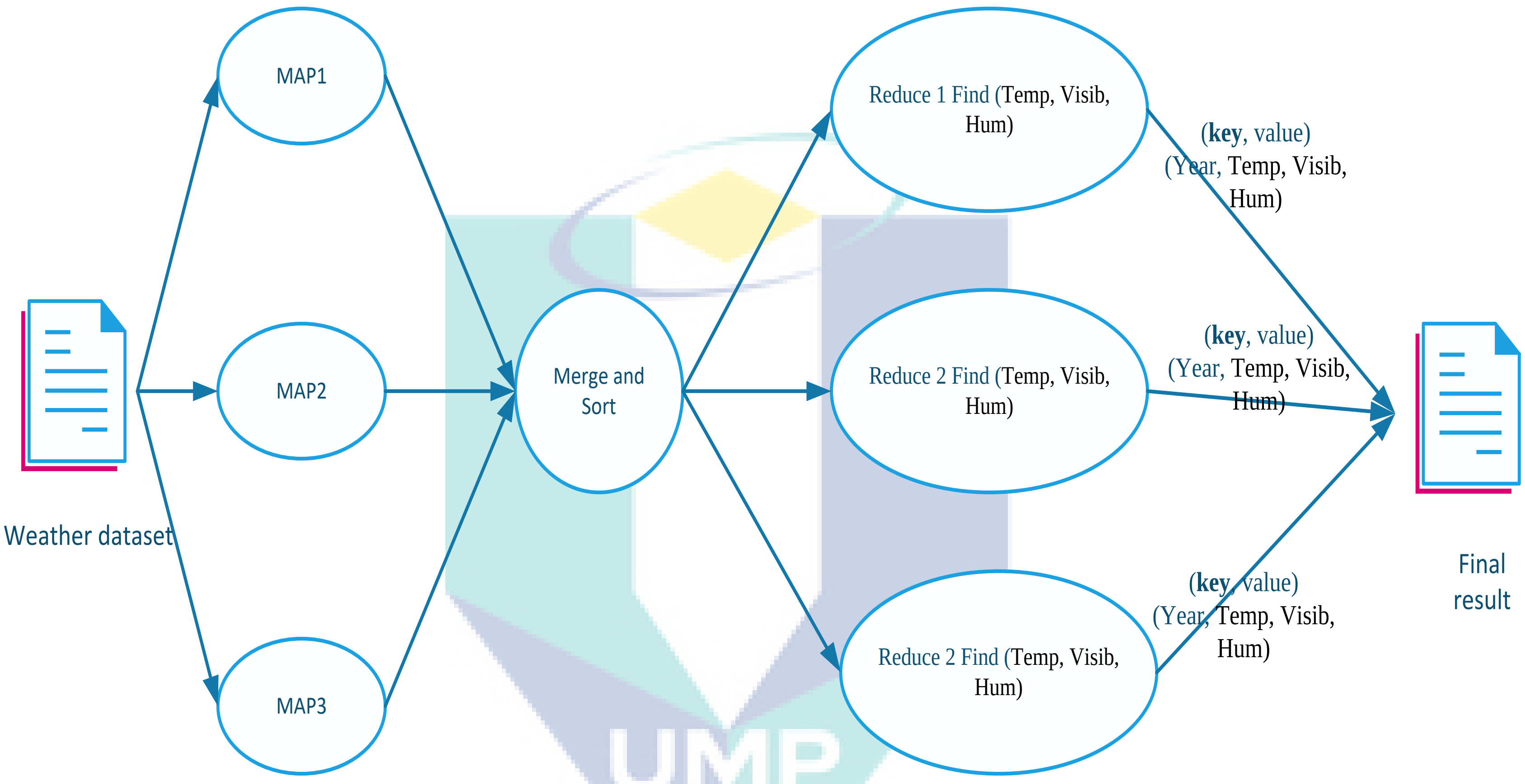


Figure 3.3. Proposed MapReduce

After the collection of dataset that we used for this study, the dataset is done in three stages; namely, map stage, shuffle stage, and reduce stage and both Map and Reduce functions operate on data conceptualized as key - value pairs.

MapReduce Algorithm comprises several components as shown in Figure 3.4 In particular, the main component is job client, which submits the job to the clusters. Job tracker oversees the task tracker and provides execution plans, coordinates the jobs, and schedules them across the task tracker. Meanwhile, task tracker breaks down the jobs into Map and Reduce tasks. Each task record has slots for execution map, gradually reduces, and reports the progress. the input data are divided into input splits based on the input format. Input splits are equated to a map task, which runs in parallel. Input format determines how the files are parsed into the MapReduce pipeline. Map transforms the input splits into intermediate key/value pairs based on the user-defined code. Shuffle and sort: moves intermediate key/value pairs outputs to the reducers and sorts them by the key. The reducer merges all pairs associated with the same (intermediate) key and then generates an output based on the user-defined code.

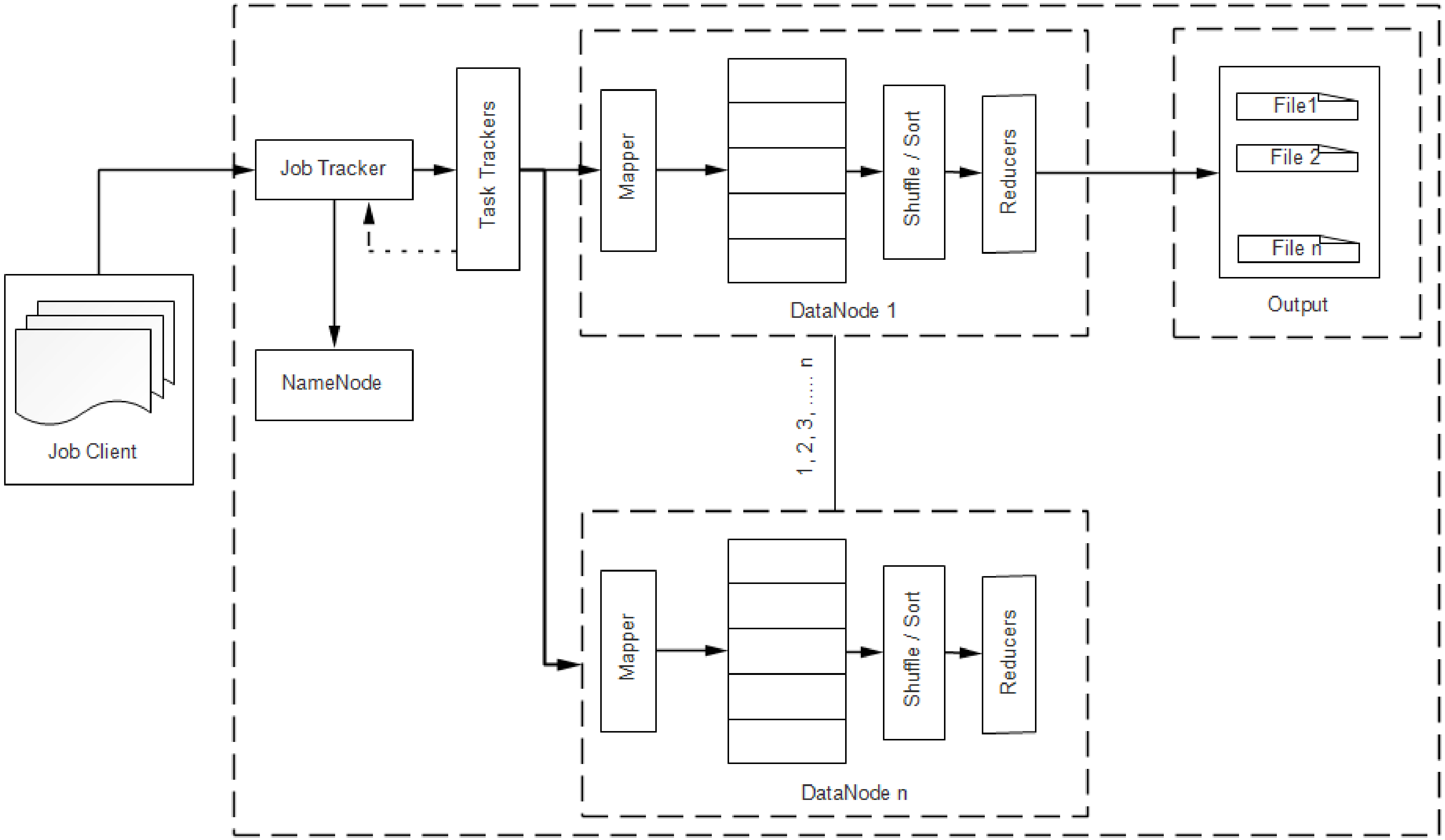


Figure 3.4. MapReduce architecture

Source: Wang et al., (2016)

The output of the MapReduce in our case is "weather dataset" year and temperature, Visibility and Humidity the map function is just a data preparation phase, setting up the data in such a way that the reducer function can do its work on it: finding the temperature, Visibility and Humidity for each year, as it is shown in Figure 3.5.

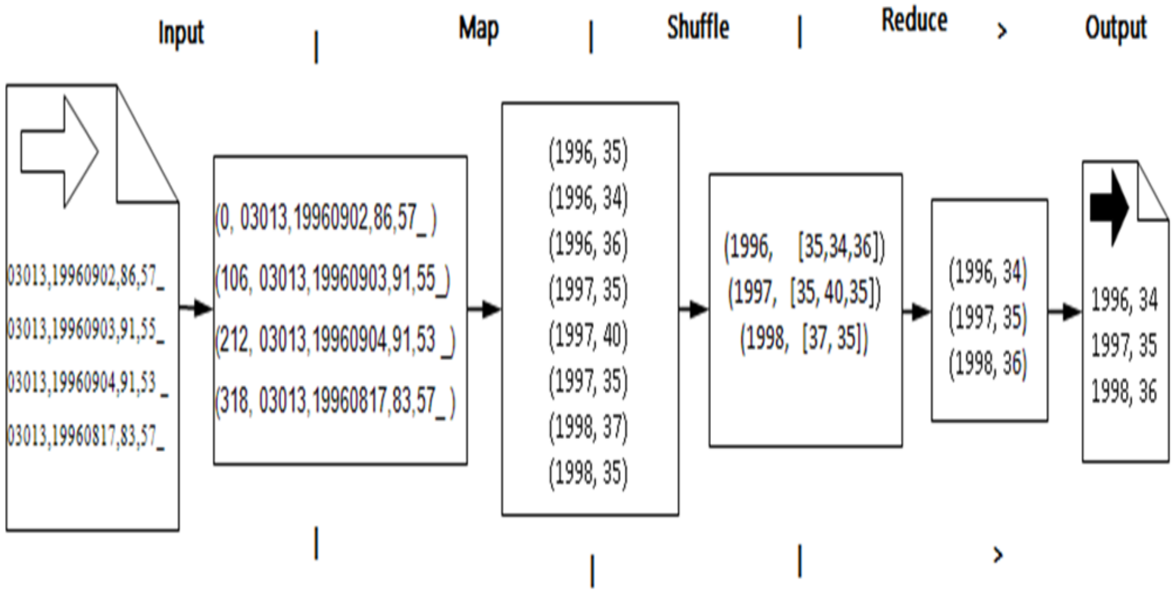


Figure 3.5. MapReduce data flow for the weather dataset

Step 1: The input weather dataset is split into a number of pieces of a specified size (128 MB). The MapReduce algorithm is started on all nodes.

Step 2: One node is set to be Master and delegating work to other Nodenodes. All pieces created in Step 1 are first mapped by the mapping function. The number of reduce tasks at the start should be low.

Step 3: If a worker gets a map task, it runs the map task and stores the result in the memory of the machine.

Step 4: Periodically these stored results are written to the disk and the Master node is notified of the action performed.

Step 5: When the Master node gets notified about the location of the mapped pairs, it will start a reduce test on one of the free workers.

Step 6: When reduce task is called, first of all, it fetches the stored results from the remote machine on which the map task has run. However, these results are sorted by key. Therefore, the results are reduced.

Step 7: When there are no more data to process, the Master node returns the final results to the user program. All this time the Master node had an overview of what all nodes are doing. The Master node will also re-assign already assigned tasks to idle nodes, because this might improve overall performance.

3.6 Experimental Setup

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop's HDFS is a highly fault-tolerant distributed file system and, like Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have Big Data sets.

The experiments were carried out in a physical cluster environment, the researcher used three computers. Hadoop cluster on Linux Ubuntu 14.04 where one computer ran a NameNode and ResourceManager and the remaining ran Datanode and DataManager. Each of the computer has the following configuration: Core i7 processor, 4 GB main memory, and 1 TB disk space as shows in Figure 3.6. The researcher used a Hadoop-2.7.1 version and the summary of Hadoop cluster in Figure 3.7. The max replication factor "dfs.replication.max" is used to set the replication limit of blocks. More details about installation Hadoop is illustrated in Appendix A.

Figure 3.6. Hadoop cluster

The master node (NameNode) web user interface shows cluster summary in Figure 3.7 including information about total/remaining capacity, live and dead nodes. Additionally, it allows to browse the HDFS namespace and view the contents of its files in the web browser. Figure 3.8. shows the information about the slaves' node (DataNodes). The default port number to access Hadoop is 50070. Use the following URL at http://master:50070/.

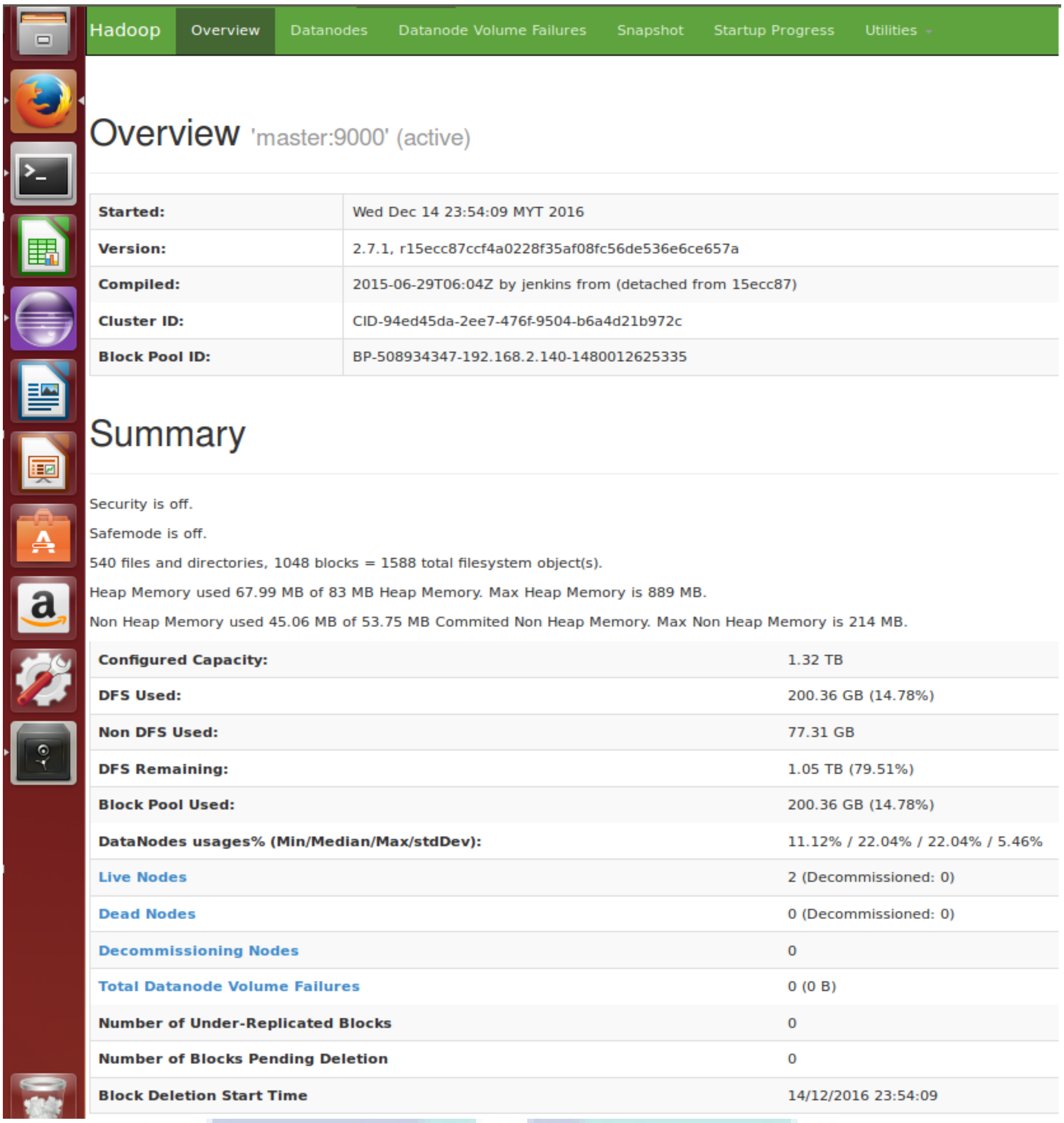
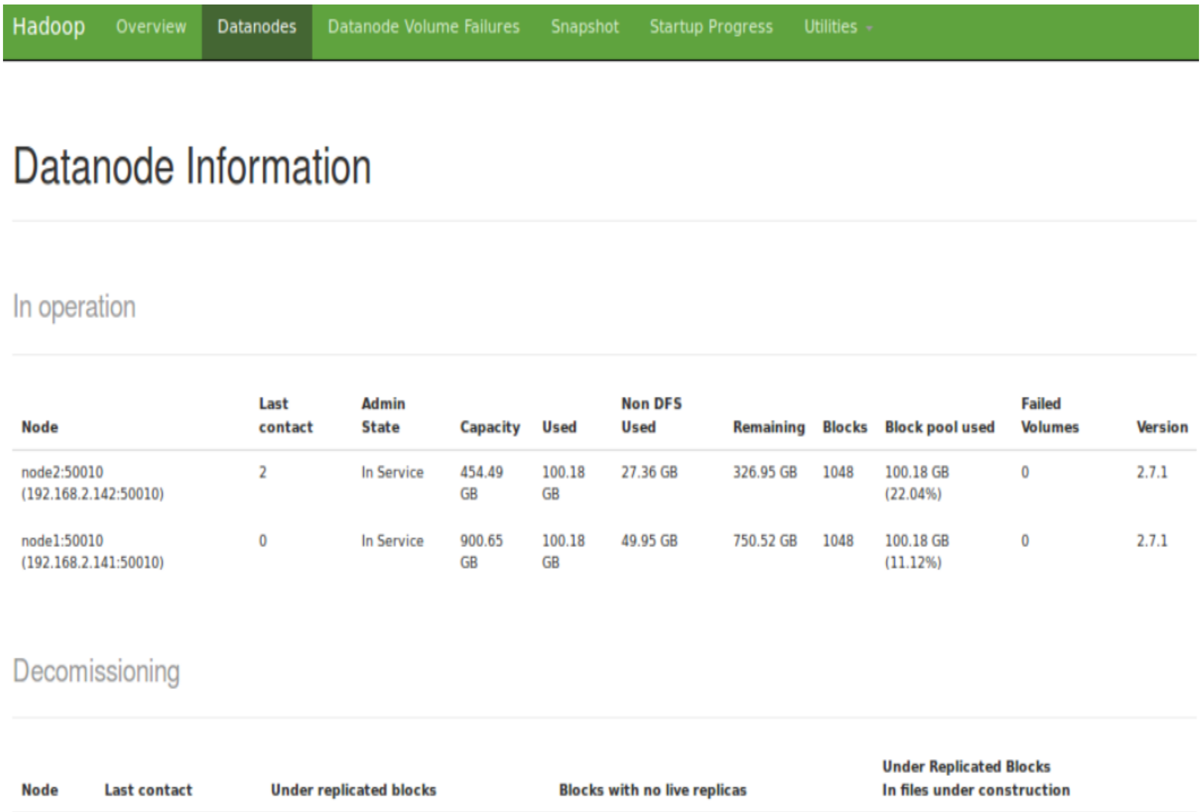


Figure 3.7. Hadoop Overview



Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
node2:50010 (192.168.2.142:50010)	2	In Service	454.49 GB	100.18 GB	27.36 GB	326.95 GB	1048	100.18 GB (22.04%)	0	2.7.1
node1:50010 (192.168.2.141:50010)	0	In Service	900.65 GB	100.18 GB	49.95 GB	750.52 GB	1048	100.18 GB (11.12%)	0	2.7.1

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction

Figure 3.8. Hadoop DataNodes Information

Figure 3.10. Node1 and Figure 3.9. Node2 show Node Manager information and that runs as separate process from WebLogic Server and allows to perform common operations tasks for a Managed Server, regardless of its location with respect to its Administration Server.

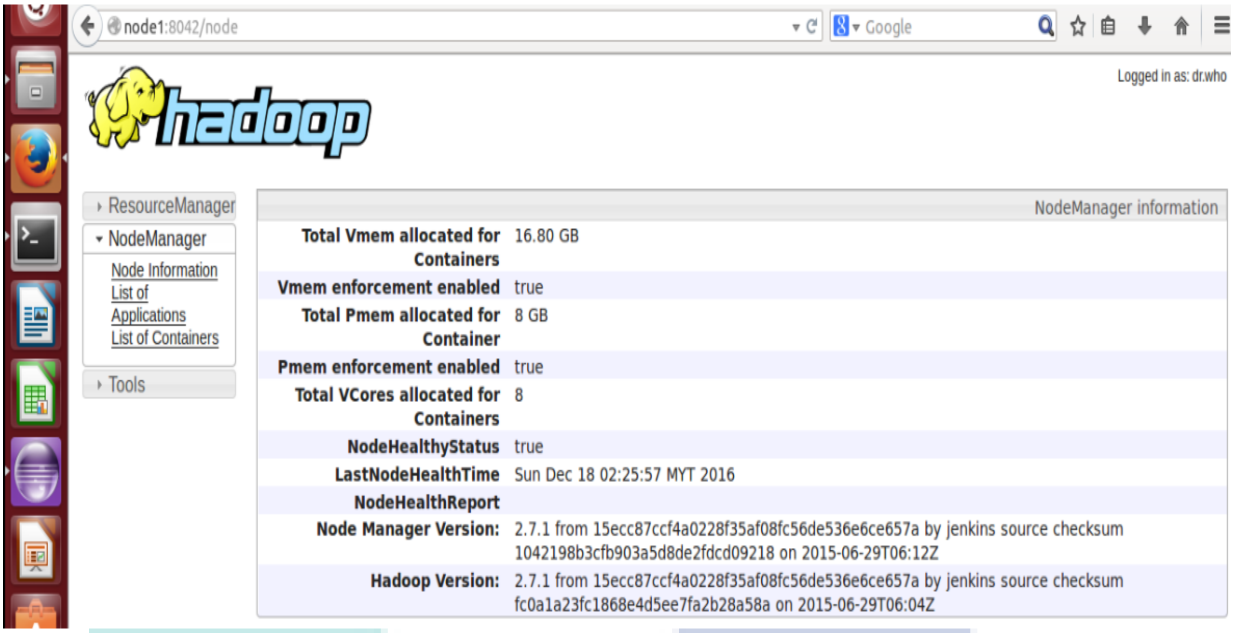


Figure 3.10. Node1- Node Manager Information

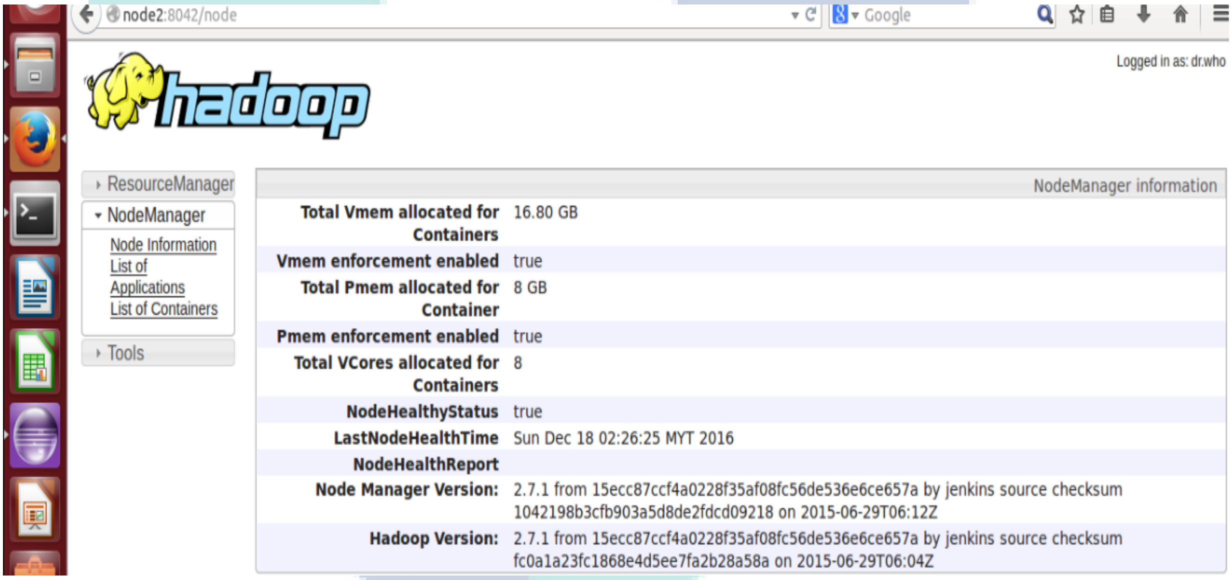


Figure 3.9. Node2- Node Manager Information

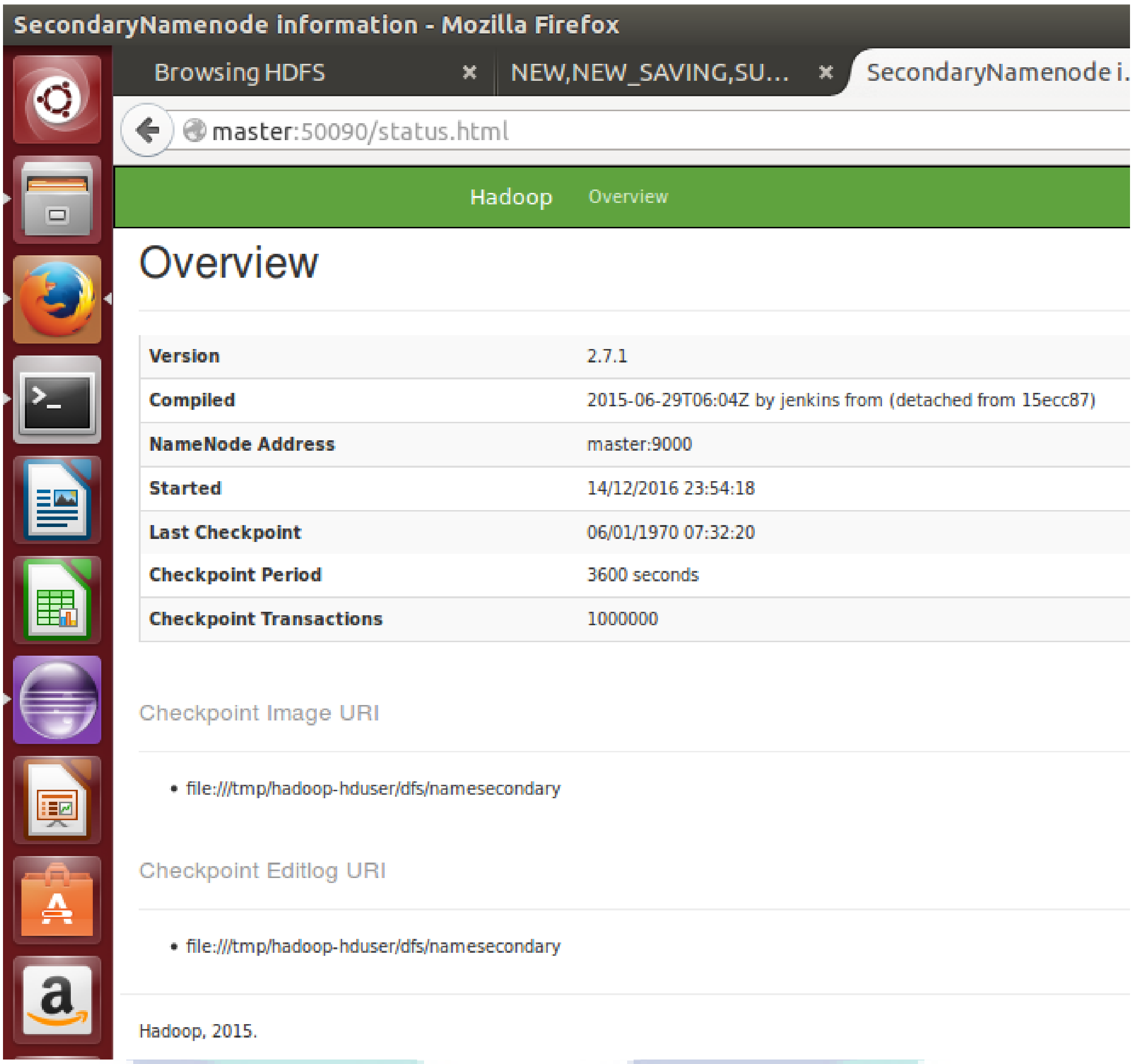


Figure 3.11. Secondary NameNode Overview

The Figure 3.11 show the Secondary NameNode is backup daemon for the NameNode. Furthermore, to evaluate the MapReduce task scheduling algorithms efficiently; the study used benchmarks representative set of CPU and IO intensive applications included in the Hadoop distribution, such as weather dataset for performance analysis in general.

3.7 Conclusions

This chapter shows the approach proposed in this study to tackle the problems raised in chapter one. The implementation of the proposed MapReduce algorithm has clearly revealed to the weather forecasting with Big Data. Further, the MapReduce algorithm establish the reliability of prepare data for prediction purposes, the proposed approach represents a robust alternative method that can be used for the weather prediction. The succeeding chapter presents the weather dataset showing MapReduce algorithm performance of the actual test run.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The results are presented in this chapter. The resulting outputs are also analyzed and comparisons are made with the results of using other methods for the same task. This was done in order to compare the proposed approach and its effectiveness in the extracting of weather dataset.

4.2 Hadoop Cluster –Data Load Performance

The cluster consists of a single NameNode (Master) server machine that manages the file system and regulates access to the filesystem by the clients. There are two DataNodes and the data is splitted into blocks and stored on these DataNodes. The NameNode maintains the map of the weather dataset distribution. DataNodes are responsible for data read and write operations during execution of weather data analysis, can also define which weather data chunks to save on which racks. This is to prevent loss of all the data if an entire rack fails and also for better network performance by avoiding having to move big chunks of bulky data across the racks. This can be achieved by spreading replicated data blocks on the machines on different racks.

4.3 Results Based on the Proposed Algorithm

The proposed algorithm extracts information from weather dataset, according to the key value of mappers and passed to reducers. The reducers do the actual processing on this reduced data provided by mappers and accomplish the final task yielding desired output. Partitioner controls the partitioning of the keys of the intermediate mapper outputs, in which the key or a subset of the key is used to derive the partition, the MapReduce algorithm has proved to be very efficient on Big Data weather dataset which is and increasingly fast.

The Hadoop job usually splits the input weather dataset into independent chunks which are processed by the map tasks in a completely parallel manner. The Hadoop sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file-system. The master node takes care of scheduling tasks, monitoring them and re-execute the failed tasks. Typically, the compute nodes and the storage nodes are the same, that is, the MapReduce and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the MapReduce to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

4.4 Execution and Results

The analysis base on MapReduce algorithm, it distributed the weather dataset in the cluster of computers (nodes). In a "map" operation the master node takes the input, partitions it into smaller sub-problems, and distributes them to data nodes. The data node processes the smaller data and passes the answer back to a reducer node to perform the reduction operation. In a "reduce" step, the reducer node then collects the answers to all the sub-problems and combines them in some way to form the output, the answer to the data it was originally trying to solve. The map and reduce functions of Map-Reduce are both defined with respect to data unfstructured in <key, value> pairs.

MapReduce process that is executed for the averaging operation on the weather dataset: The weather dataset files were processed into Hadoop sequence files on the HDFS master Node. The files were read from the local weather directory, sequenced, and written back out to a local disk. The resulting sequence files were then ingested into the Hadoop file system with the default replica factor of three and, initially, the default block size of 64 MB. The job containing the actual MapReduce operation was submitted to the master Node to be run. Along with the JobTracker, the Master Node schedules and runs the job on the cluster. Hadoop distributes all the mappers across all data nodes that contain the data to be analyzed. On each data node, the input format reader opens up each sequence file for reading and passes all the <key,value> pairs to the mapping function.

Map Phase: The input for Map phase is set of weather dataset files as shown in snap shot Figure 4.1. The types of input key value pairs are LongWritable and Text and the types of output key value pairs are Text and IntWritable. Each Map task extracts the temperature, Visibility and Humidity from the given year file. The output of the map phase is set of key value pairs. Set of keys are the years. Values are the temperature, Visibility and Humidity of each year.

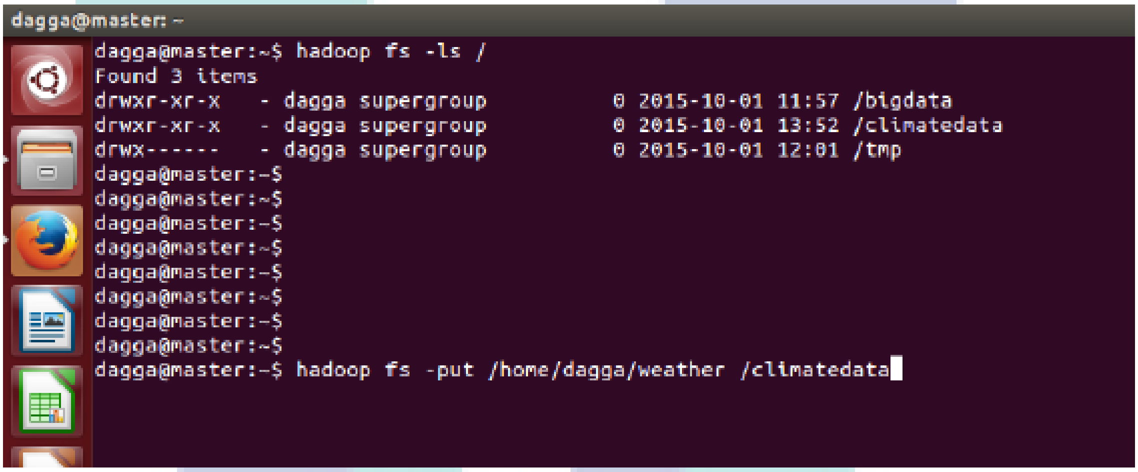


Figure 4.1. Push the dataset into climatedata folder

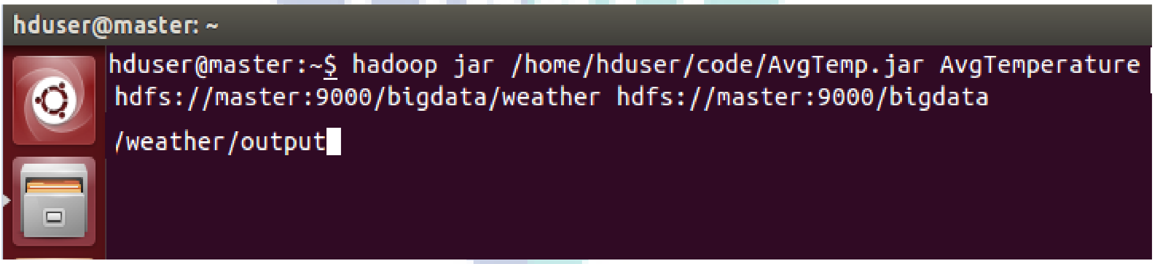


Figure 4.2. Run the weather dataset

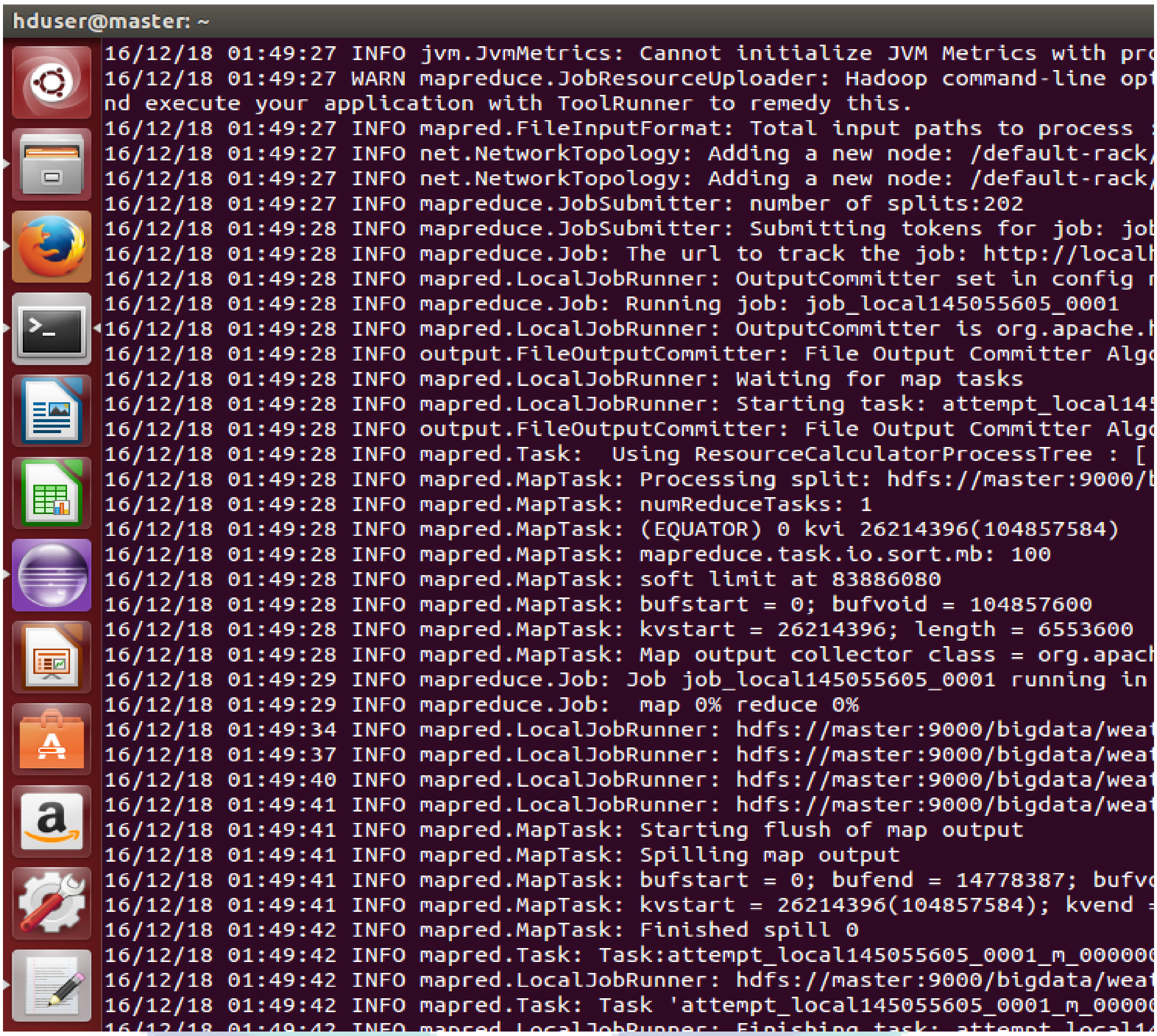


Figure 4.3. The process of MapReduce

Reduce Phase: Reduce phase takes all the values associated with a particular key. That is all the temperature values belong to a particular year is fed to a same reducer. Then each reducer finds the average recorded temperature for each year. The types of output key value pairs in Map phase is same for the types of input key value pairs in reduce phase (Text and IntWritable). The types of output key value pairs in reduce phase is to Text and IntWritable.

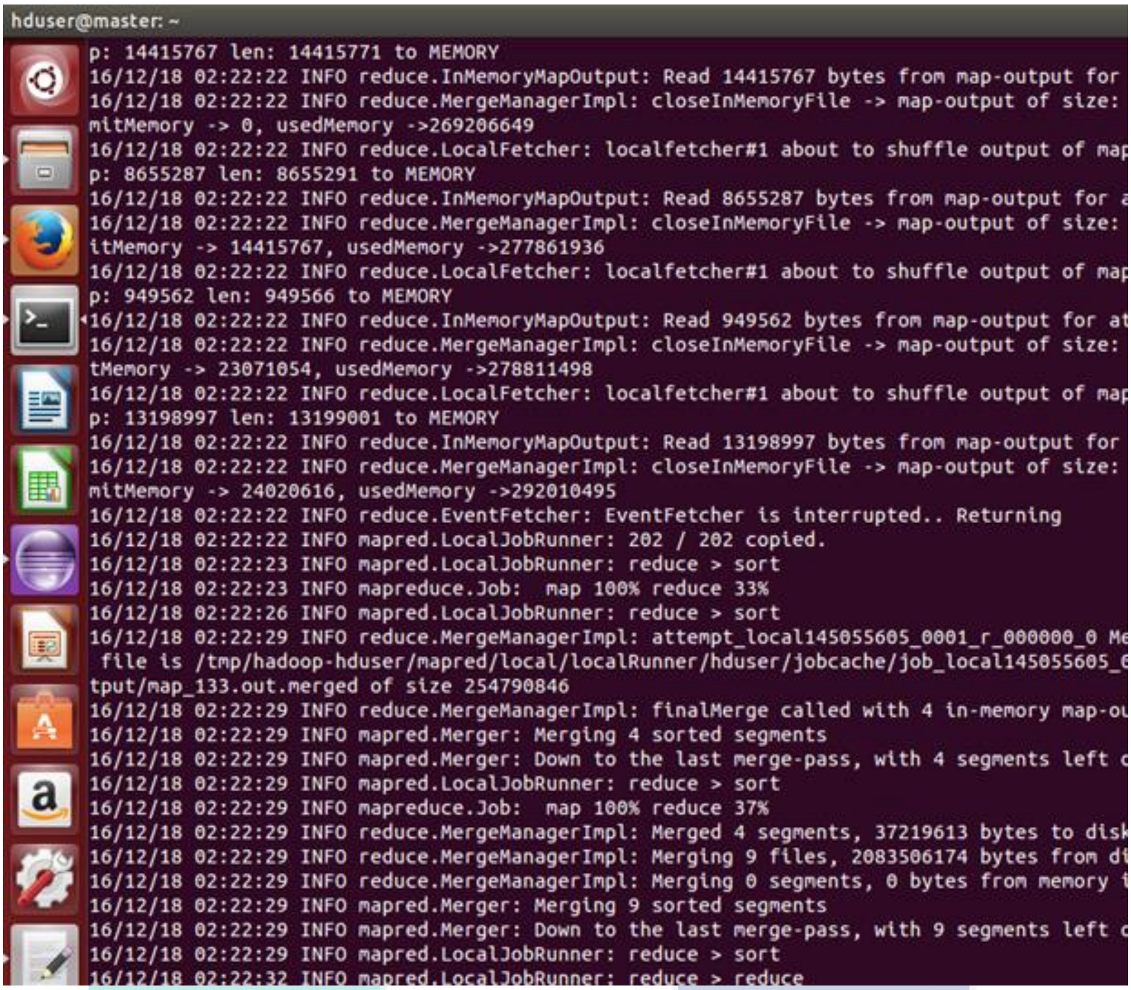


Figure 4.4. The process of Reduce

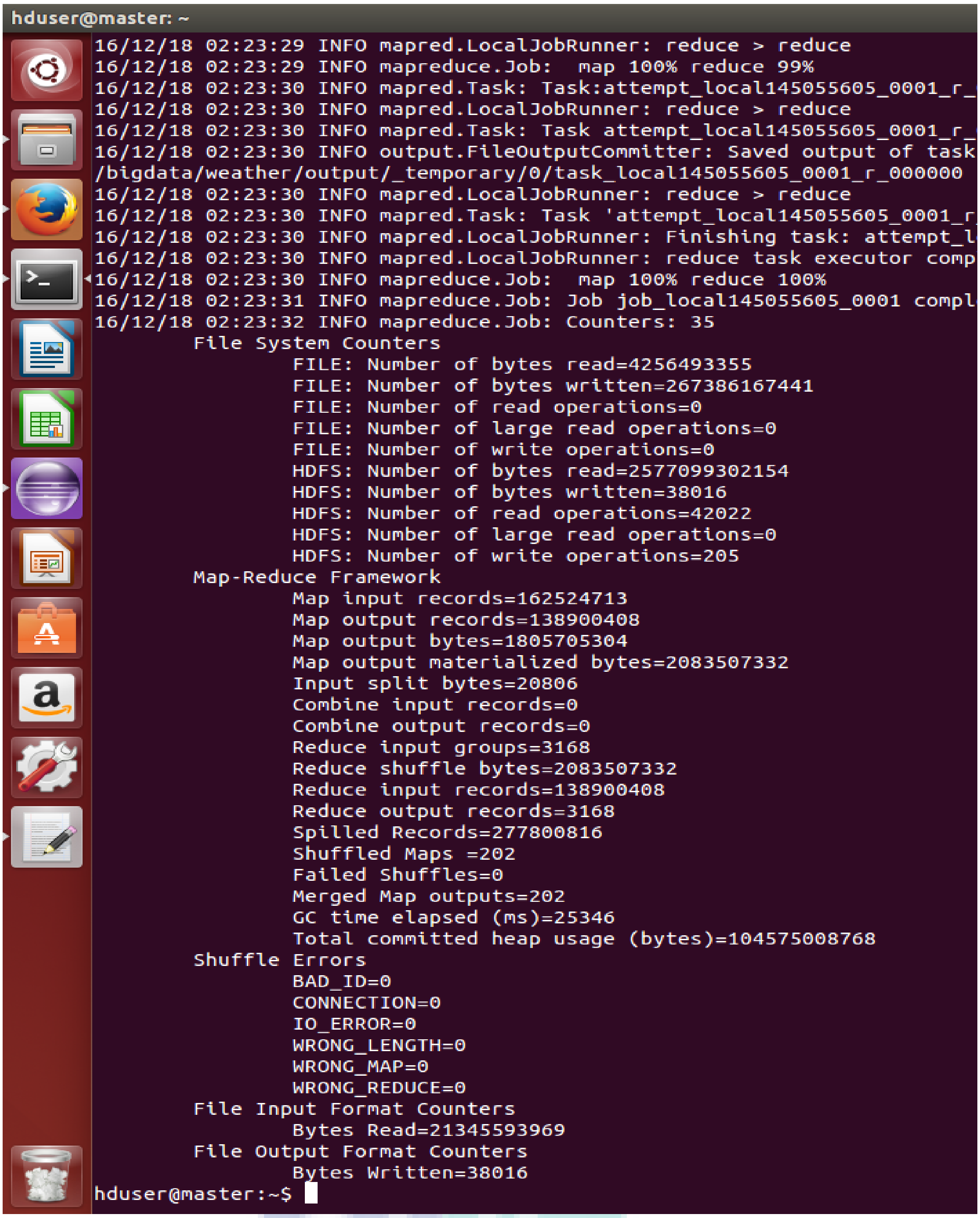


Figure 4.5 (a). The Output of MapReduce process

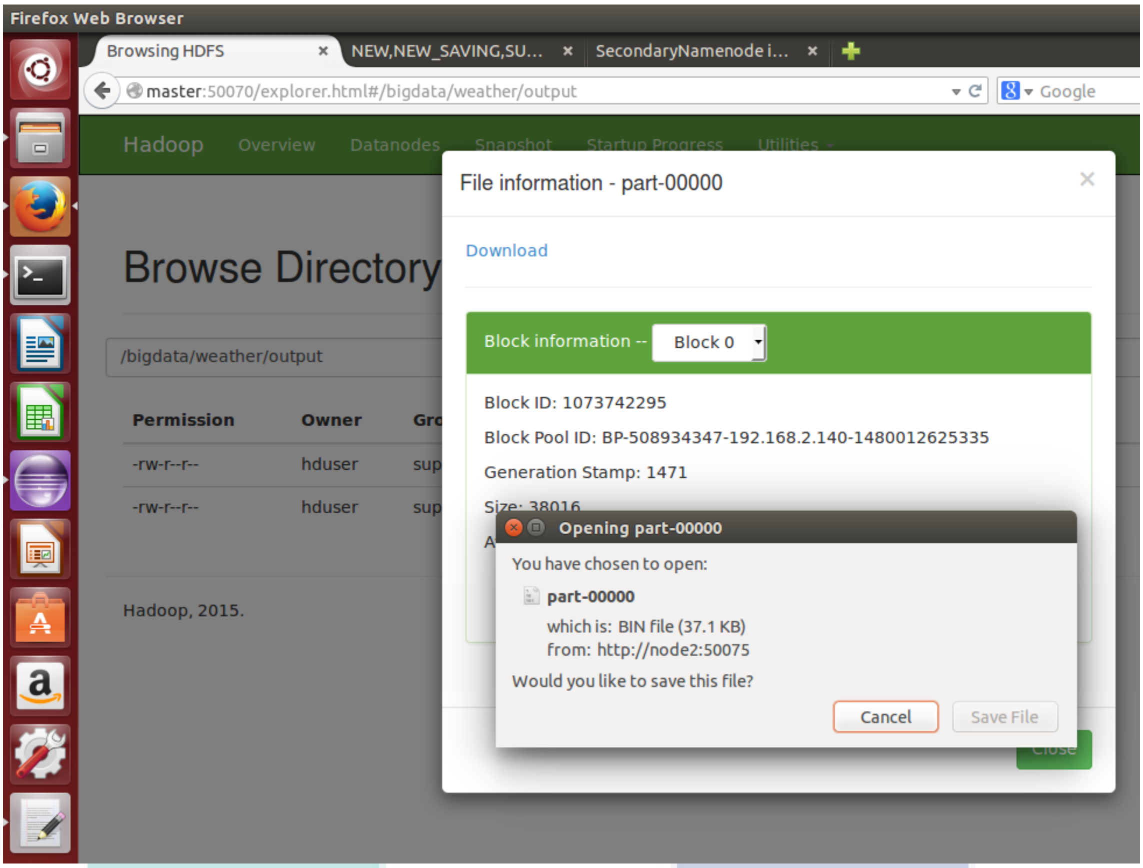


Figure 4.5 (b). The Output of MapReduce process

It can easily be seen from Figures Figure 4.5 (a) that query execution time significantly goes down with increased number of nodes. This is due to the fact that more number of cores is available to execute the MapReduce tasks in parallel. Since Hadoop brings computation to nodes, there isn't any overhead involved in transferring the data to be analysed.

4.4.1 Experiment Results

The researcher study weather dataset from the National Oceanic and Atmospheric Administration (NOAA) over a period of 10 years. The data are collected for the years from 1997 to 2007. In some cases, data were missing from some weather stations. In the following paragraphs, we present our analysis of the collected data. Due to size limitations, we only show the Figures with significant interest.

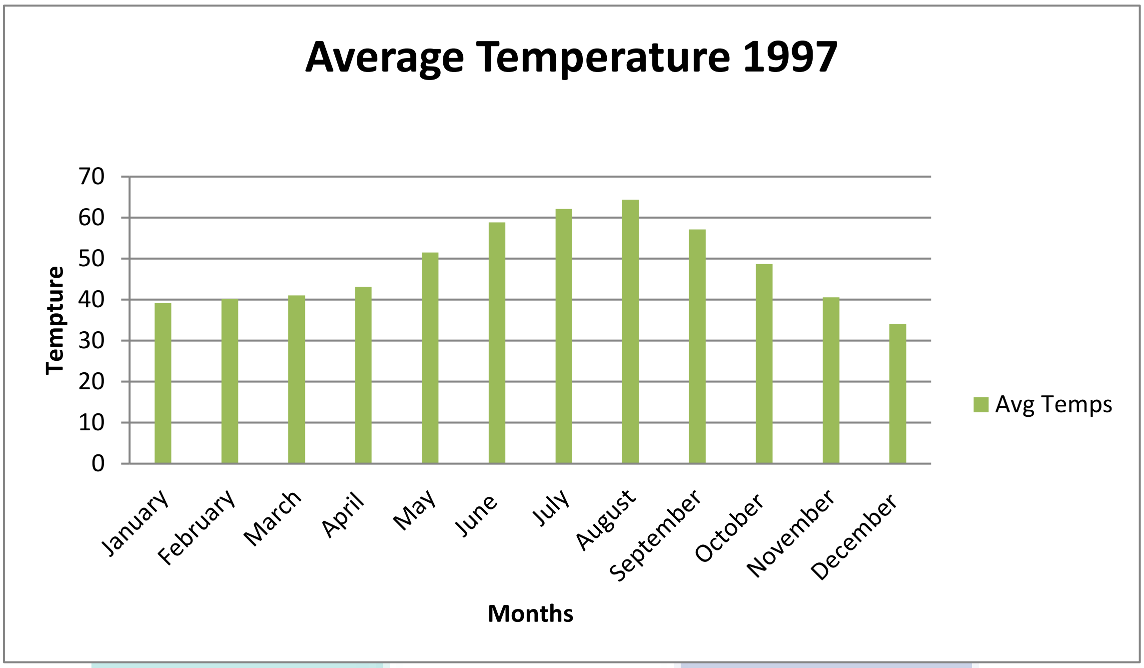


Figure 4.6 (a). Average Temperatures in 1997

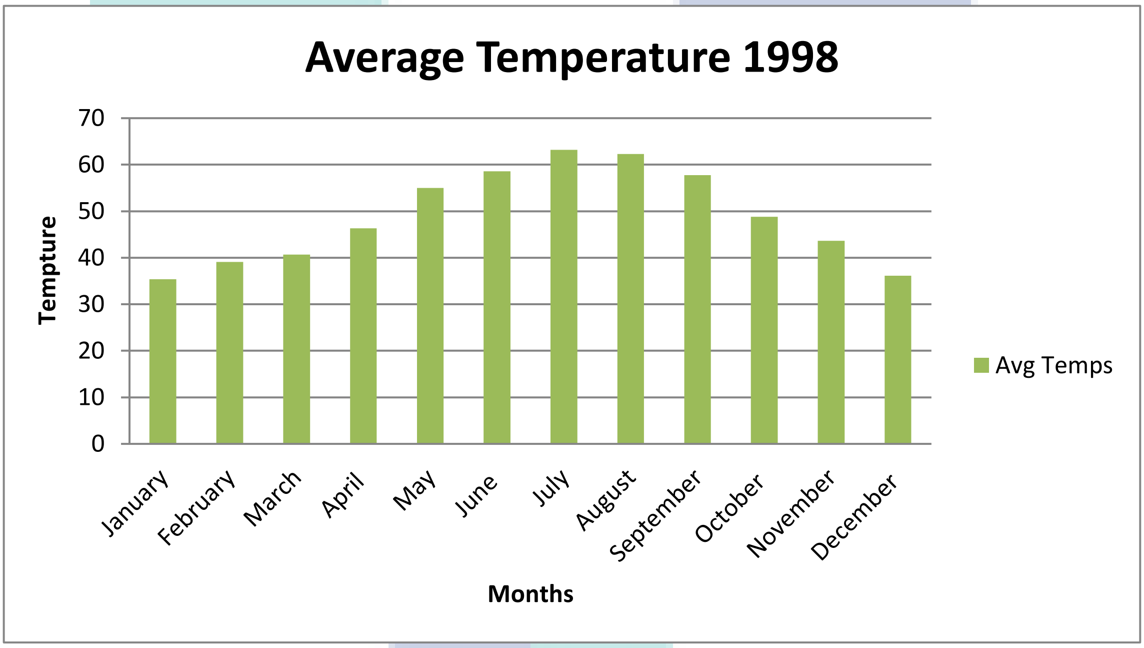


Figure 4.6 (b). Average Temperatures in 1998

Figure 4.6 (a) and (b) show yearly evolution for weather dataset. We just selected the temperature factor to reduce the clutter in the picture and improve visibility. Fluctuations of the average temperatures vary from (41- 45 °C) or (50 - 62°C) down to 35 or 32 °C (July and August shows the highest average temperature). Low degrees are largely on December.

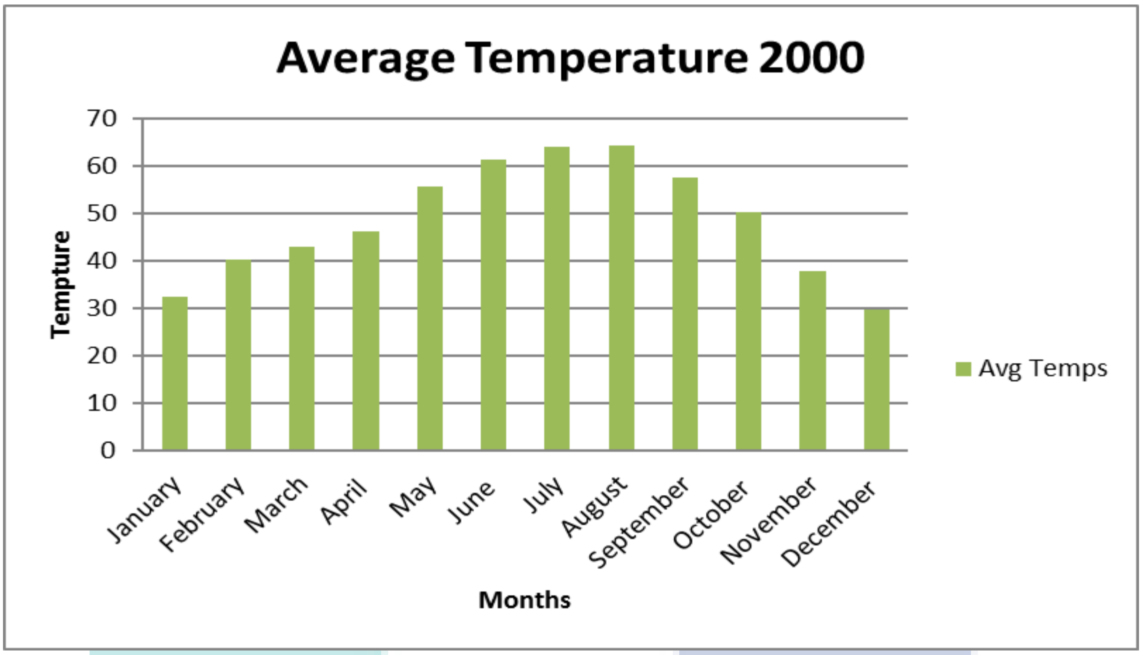


Figure 4.7. Average Temperatures in 2000

Figure 4.7 shows the Average Temperatures trend from January to December. It is observing that the Average Temperatures from January to July increase gradually. Meanwhile dropped slightly from August to December.

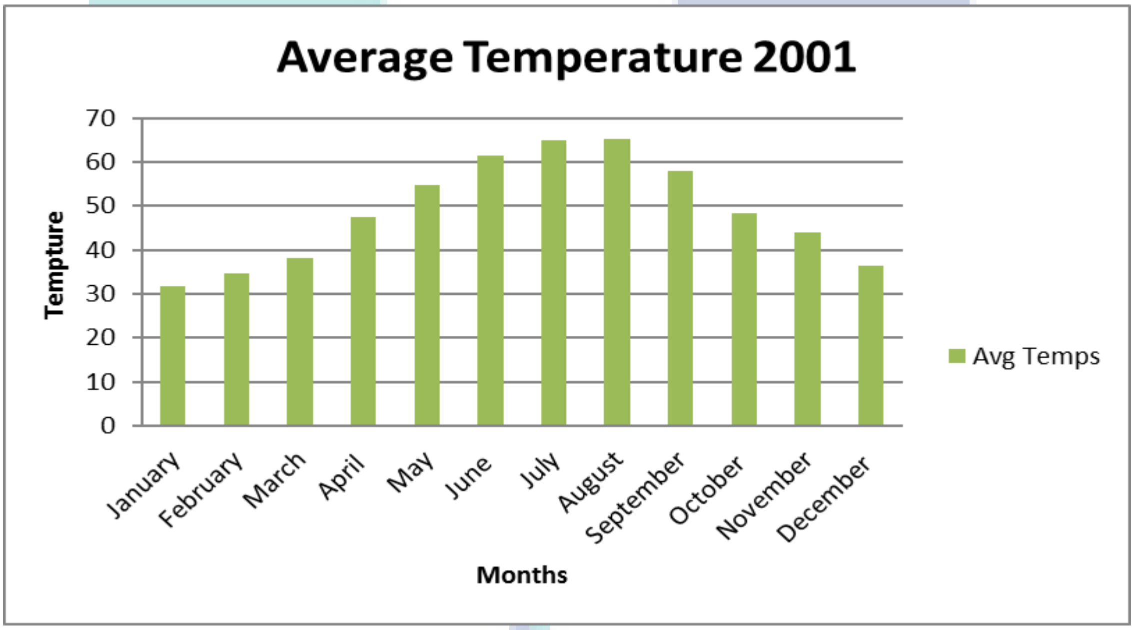


Figure 4.8. Average Temperatures in 2001

Figure 4.8 shows the Average Temperatures trend from January to December. It is observing that the Average Temperatures from January to June increase gradually. Meanwhile, almost similar between July to August. In addition, dropped slightly from September to December.

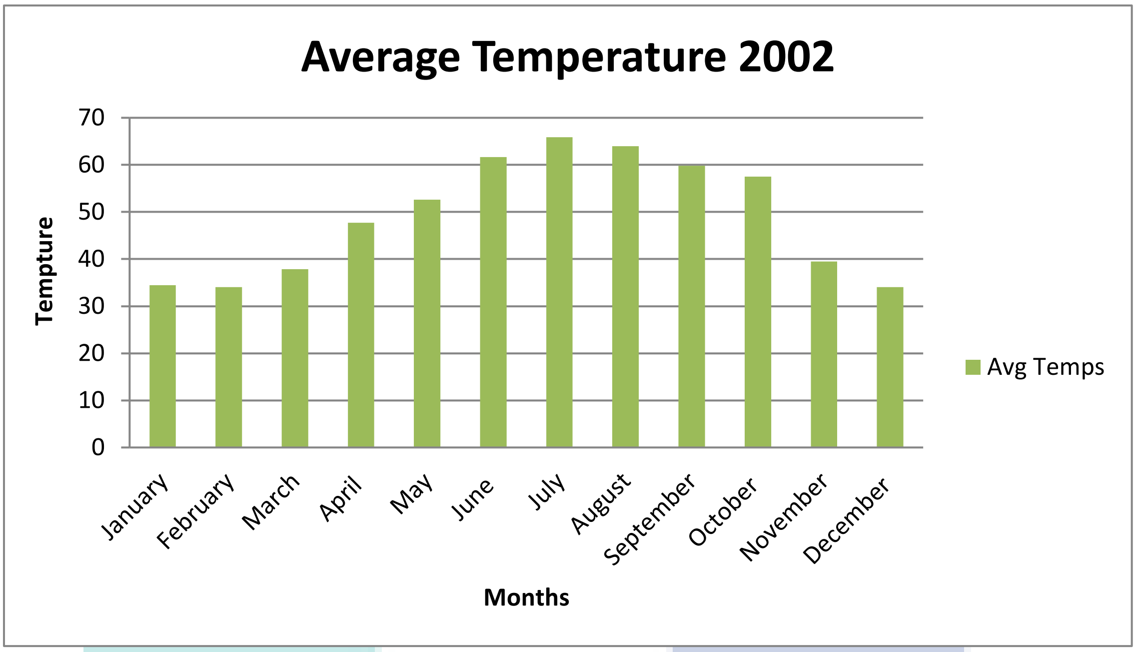


Figure 4.9. Average Temperatures in 2002

Figure 4.9 shows the Average Temperatures trend from January to December. It is observing that the Average Temperatures from January and February are similar. Meanwhile increase gradually from March to July. In addition, the lower in December.

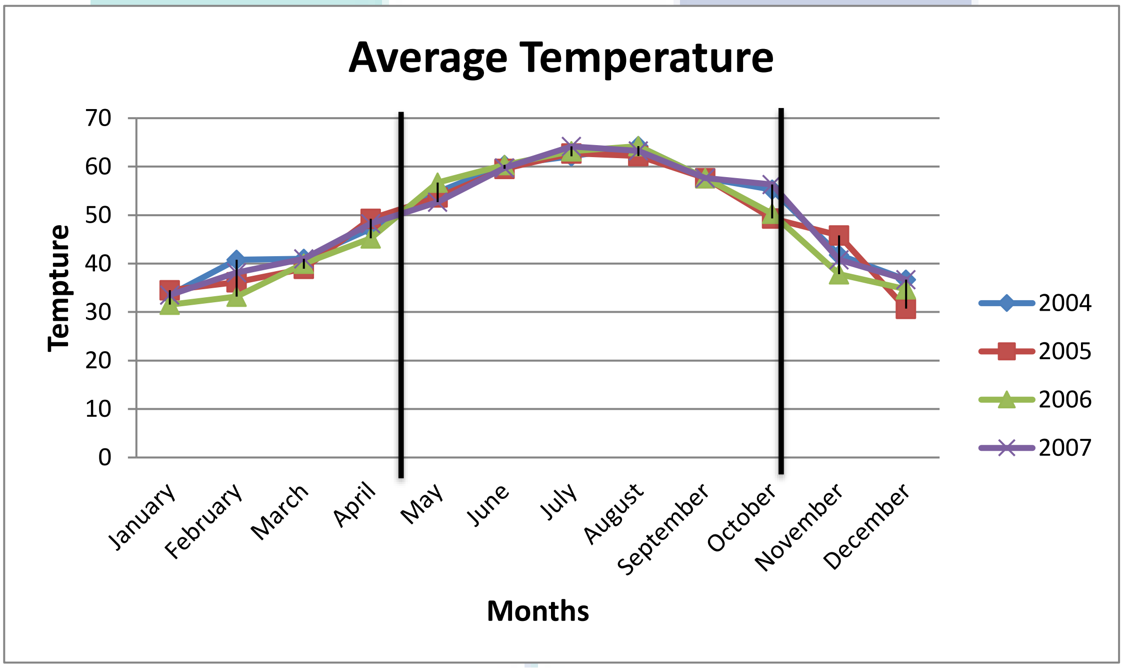


Figure 4.10. Average Temperatures for four years

Weather forecasting statistics Figure 4:10 indicates the annual pattern of weather forecasting function of months of the year for the last 10 years. Note that most of the high average temperature weather forecasting occur during the convective season (May, Jun, Jul, Aug, Sep and Oct).

The Figure 4.11, Figure 4.12 and Figure 4.13 shows the relative humidity ranges from comfortable to very humid over the year. rarely dropping to very dry and reaching very humid for some months.

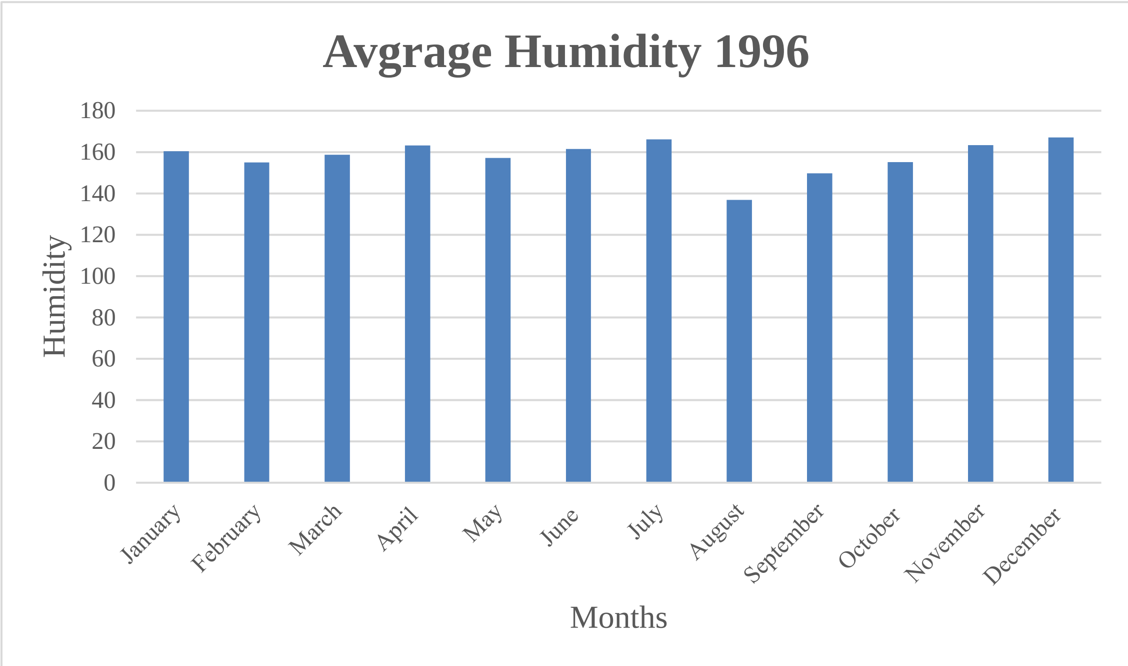


Figure 4.11. Average Humidity in 2000

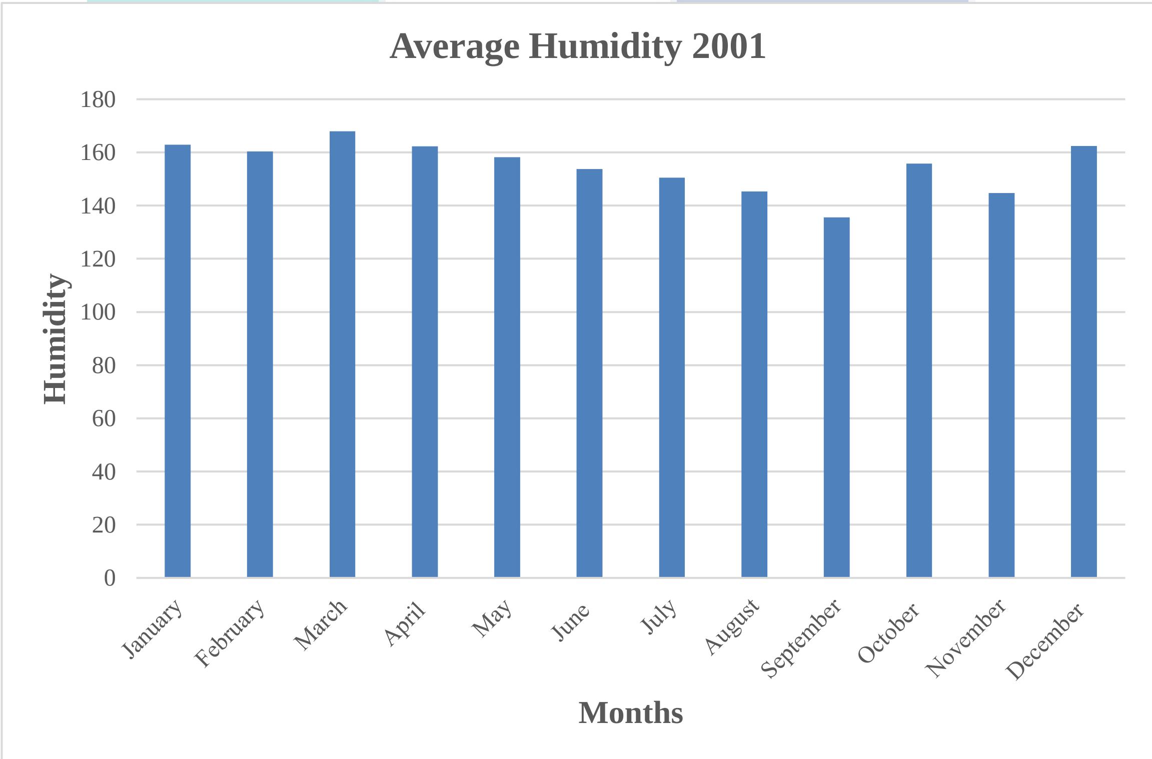


Figure 4.12. Average Humidity in 2001

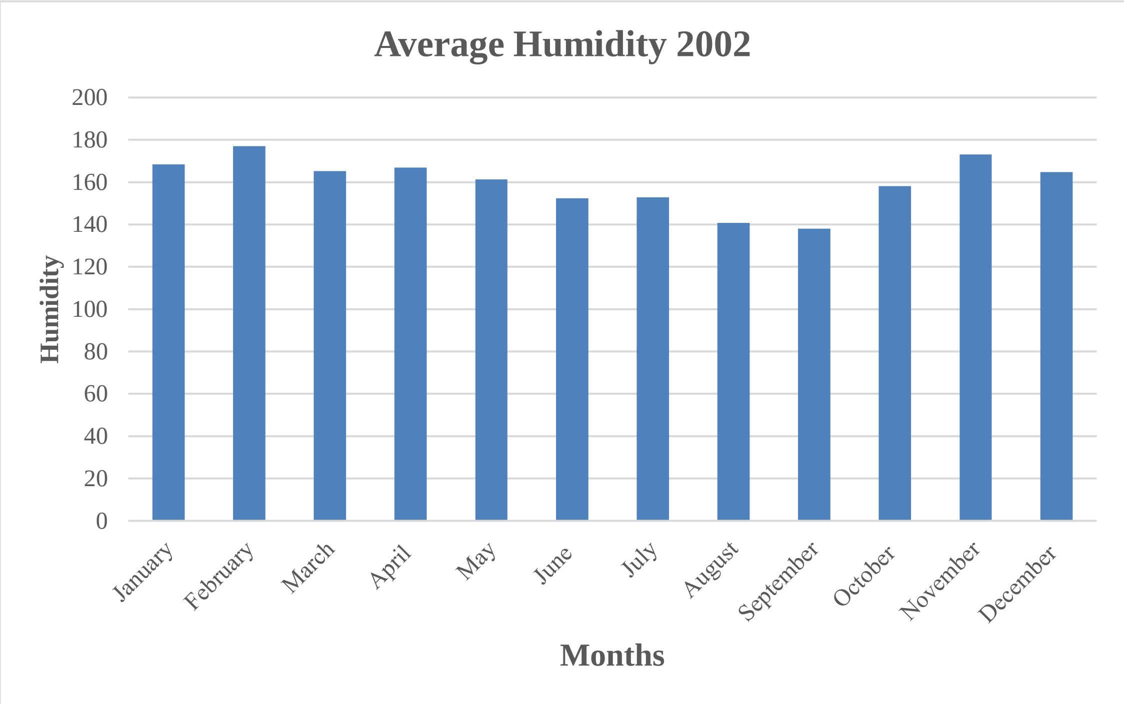


Figure 4.13. Average Humidity in 2002

Visibility is a measure of the distance at which an object or light can be clearly discerned. Visibility affects all forms of traffic: roads, sailing and aviation. Meteorological visibility refers to the transparency of air and in dark. Figure 4.14, Figure 4.15 and Figure 4.16 show the average visibility.

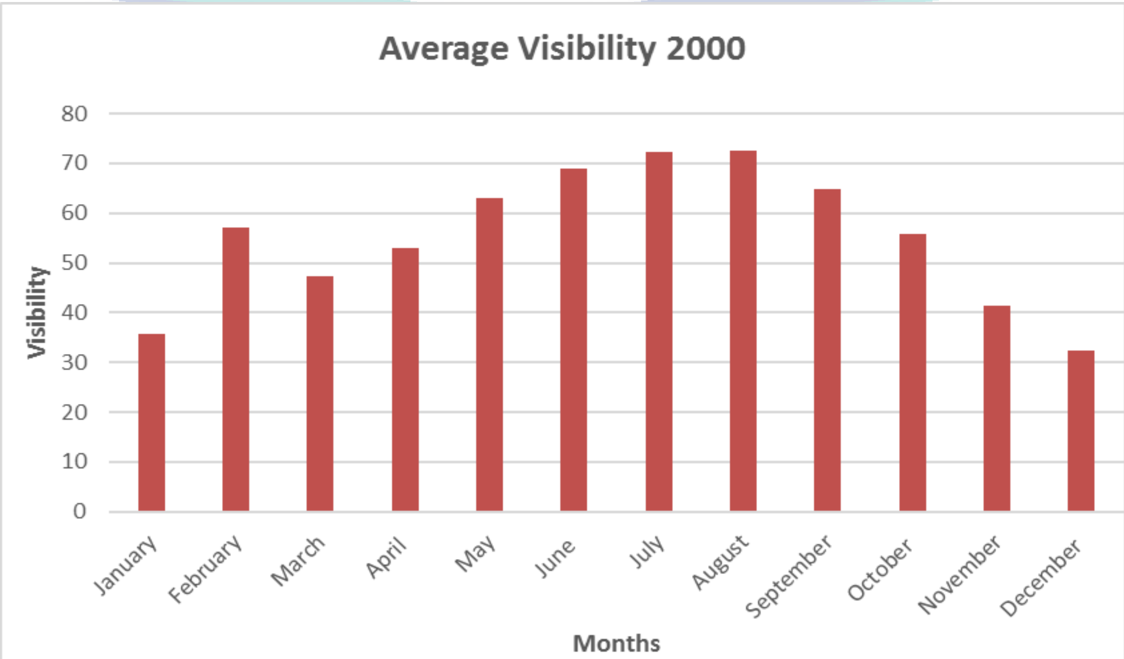


Figure 4.14. Average Visibility in 2000

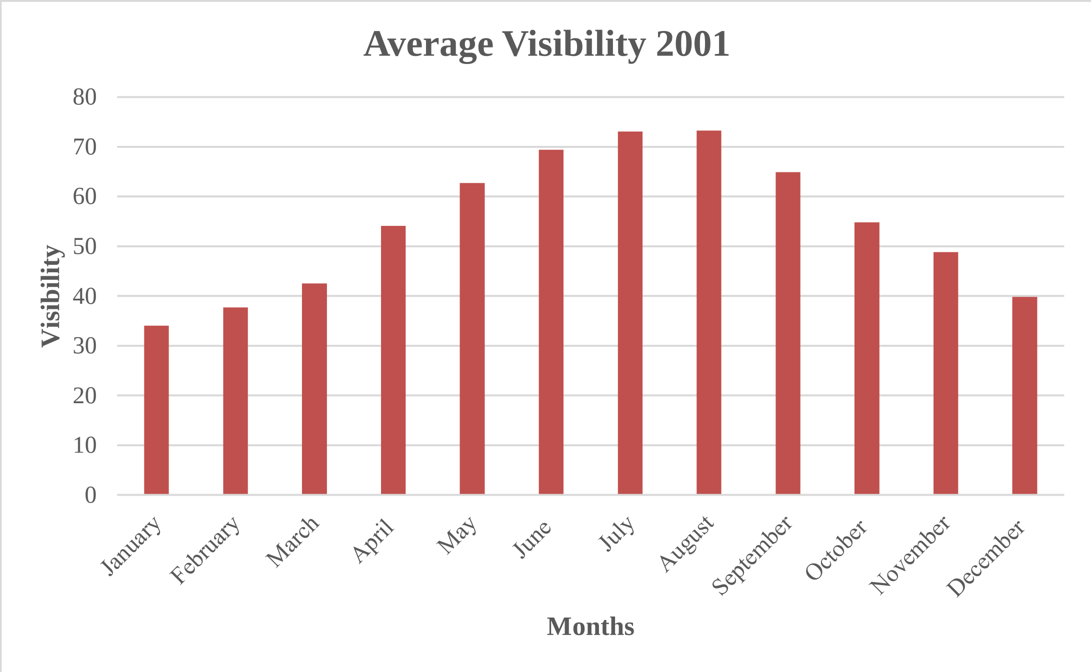


Figure 4.15. Average Visibility in 2001

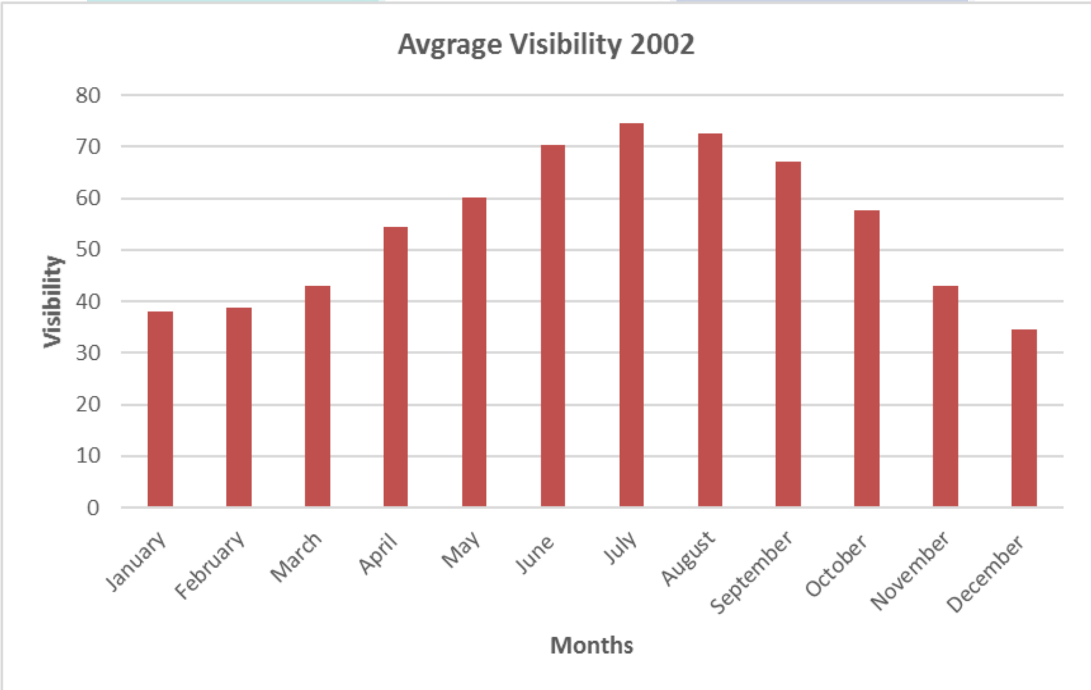


Figure 4.16. Average Visibility in 2002

4.5 Comparison of the Proposed Approach

In this study, the proposed MapReduce algorithm is used for weather data process. The temperature, visibility and humidity it extracted based on the MapReduce algorithm. Through the implementation of the algorithm, it is illustrated, how MapReduce algorithm efficiently integrated with big weather dataset. This algorithm proves simplified gradient algorithm. Moreover, the implement of the MapReduce algorithm in the cluster have high performance. And there not a substantial number of errors in data processing. Hence the proposed approach extracted the temperature, visibility and humidity it is capable of yielding good results and can be considered as an alternative to traditional meteorological approaches. This approach is able to determine the weather prediction more accuracy in future. Additionally, data mining algorithms are suggested an important method which could be used for weather forecasting with Big Data. But these techniques have been designed to handle data which are comparatively smaller sizes as opposed to the size of Big Data.

Likewise, AWK (**A**ho, **W**einberger, and **K**ernighan) model is designed for text processing and typically used as a data extraction. It is an excellent tool for processing the weather data and it is easier to use than most conventional programming languages. But AWK provides low-level implementation in terms of time process, parallel process and combine results compared to MapReduce as follow.

First, dividing the work into equal-size pieces isn't always easy. The weather data from (NOAA), it from different years varies widely and different size of the data files. In this case, if use AWK, some of processes will finish much earlier than others. Thus, the time process for any task runs, it's dominated by the longest data file. As a result, the proposed approach divides the weather data into fixed-size chunks and assigns each chunk to a process. So it reduces the processing time.

Second, combining results from independent processes need more processing in AWK. Because the result for each year is independent of other years. Therefore, the results are compiled and sorting by year.

In MapReduce approach used the fixed-size data chunk and that make the combination is more accurate. The final step is looking for the average temperature, visibility and humidity for each year.

Third, in AWK the capacity of a single machine is limited. And the weather dataset is big beyond the capacity of a single machine. Thus, the time processing in the single machine it long. Compared with parallel processing, which takes a short time to processing in MapReduce.

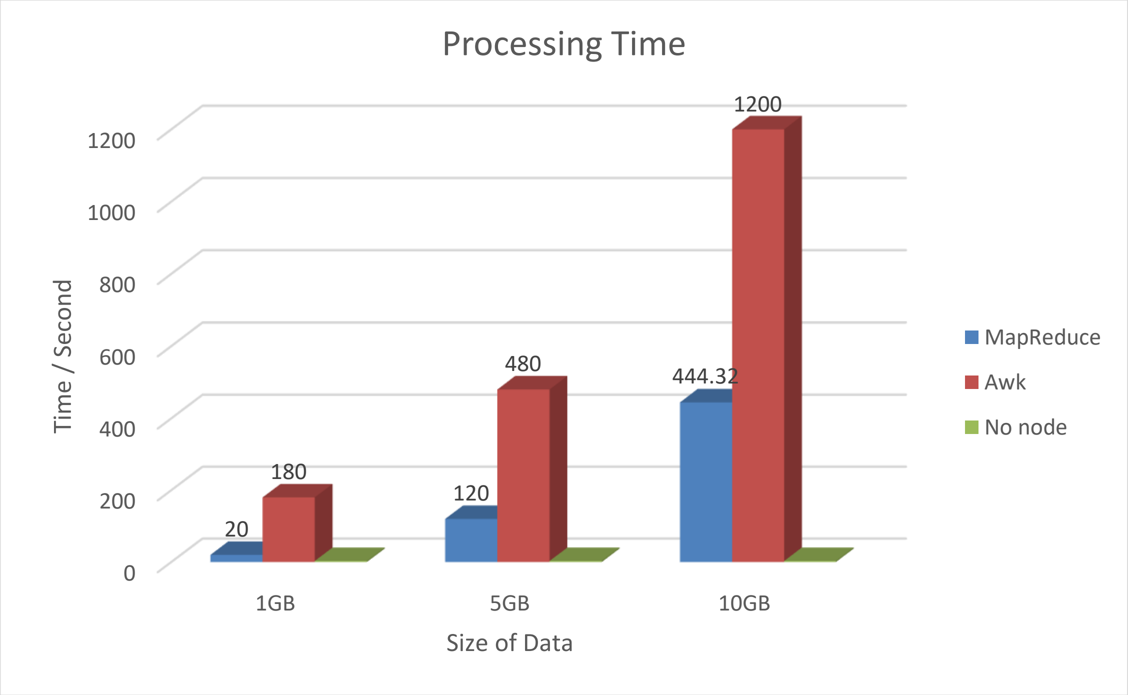


Figure 4:17. Comparison of execution time with different size dataset

For this research we have taken weather data form NOAA. These data unsaturated contain (temperature, wind speed, humidity, etc.). We have installed Hadoop/Map Reduce programming paradigm. The Analyzed result shows that the MapReduce has a good scalability and stability for extract value from weather Big Data. We have also taken result of AWK programming uses the same data for comparison with the MapReduce as shows in Figure 4:17. The execution time of the MapReduce in 10 GB (37%), 5 GB (25%), 1 GB (11%) is less compare to AWK model.

Table 4.1 Comparison of MapReduce to other methods

Techniques	**Metrics**			
	Capable to parallel processing	**Small dataset**	**The Implementation allows missing value**	**Tolerance with data**
MapReduce	✓	x	✓	✓
AWK	x	✓	✓	x

Note: ✓ supported

X not supported

4.6 Conclusion

This chapter shows the results of the MapReduce algorithm. The results are illustrated in graphical formats. Based on the metrics under consideration, the MapReduce algorithm shows some high degree of reliability for Big Data weather dataset. This was further confirmed when the technique of Big Data for weather forecasting result.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter concludes this thesis. The approach that is proposed for the Big Data weather dataset is summarized. The discussion here gives answer to the problem statements earlier listed in Chapter 1. Also, discussed in this chapter are the rationales for weather forecasting with Big Data and how the proposed approach has contributed to knowledge is further discussed in this chapter. Some recommendations are also suggested regarding how the present study can be extended in the near future.

5.2 Summary

In traditional algorithms, the processing of millions of records is the time-consuming process. In the era of Big Data, the meteorological department uses different sensors to get the temperature, humidity etc. values. MapReduce is programming model for executing highly parallelizable and distributable algorithms across big datasets using a large number of commodity computers. Weather forecasting plays a vital role in human daily routine, business and their decisions. The technology for weather forecasting is evolving rapidly due to the critical needs in obtaining the accurate prediction results. From the literature exploration, the researchers have found that weather data is important to be analysed in form of structure data. Most of data in weather is represented in unstructured data with different attributes such as temperature, humidity and visibility.

This study presents, MapReduce algorithm for weather dataset. The research is focus on analyzing the weather dataset using MapReduce Algorithm. The historical dataset in 10 years' period (1997 to 2007) has been used and this dataset is obtained from NOAA. Therefore, we extract features from a weather dataset and use the temperature, visibility and humidity factors. The experiment result shows that the proposed MapReduce algorithm is to investigate the capability of the proposed model in parallel processing. The comparison results shown that MapReduce Algorithm has produced 37%, 25% and 11% less compared to AWK in term of processing time for 10GB, 5GB and 1GB data, respectively. This result has revealed the significant impact to the used of MapReduce Algorithm in weather prediction. The scalability bottleneck is removed by using Hadoop/MapReduce. Moreover, the Hadoop/MapReduce distributed network gives faster processing of the data. With the widespread employment of these technologies throughout the commercial industry and the interests of the open-source communities, the capabilities of MapReduce and Hadoop will continue to grow and mature. The use of these types of technologies for Big Data analyses has the potential to greatly enhance the weather forecast too.

5.3 Limitations of This Study

Although, the intent of the MapReduce algorithm to weather forecasting with Big Data, the approach for temperature, visibility and humidity factors is capable of yielding good results and can be considered as an alternative to traditional meteorological approaches. The limitation of this study proposed only uses the unstructured data instead of using both (structured data and sim-structured data) for efficiency temperature, visibility and humidity prediction accuracy.

5.4 Contributions to Knowledge

This research has contributed to knowledge in MapReduce algorithm for weather dataset. The specific contributions of this research are:

This research proposed MapReduce algorithm for weather dataset, which is found to be effective in weather with Big Data. The algorithm proposed is tested and comparison is made with some algorithm that well reported for modelling of data for predictive purposes. Findings show that, the proposed MapReduce algorithm is efficient and can be used for big weather dataset.

The approaches proposed in this research, has shown how weather dataset using the techniques of Big Data analysis. The techniques proposed have been able to extract Big Data weather historical dataset. The weather forecasting can be of benefit in any aspect of our life such as decision making. However, the proposed MapReduce algorithm in this study was proposed to deal with big weather dataset irrespective of the size of dataset involved.

5.5 Recommendations for Future Research

The proposed approach has shown how the weather forecasting with Big Data can be more accurate. Further research will be conducted to create a more weather forecasting prediction model for both structured, sim-structured and unstructured compared to the one that was used in this study. Such an approach would result in increasing the size of the input data, and as analysis and prediction need to be performed in each cell, the frequency of data processing and operation would increase. To address this issue, a Big Data analysis solution that can support distributed parallel processing, such as RHive, can be used.

REFERENCES

Arribas-Bel, D. (2014). Accidental, Open and Everywhere: Emerging Data Sources for the Understanding of Cities. *Applied Geography*, 49 (5), 45-53.

Avneesh T. & Rishabh S. (2015). Web Log Mining Using MapReduce and Apache Spark. *A Peer Reviewed International Journal*, 3 (5), 2321-7758.

Aster, D. & Beijing, S. (2013). Big Data Storage and Challenges. *International Journal of Computer Science and Information Technologies*, 5(2), 2218-2223.

Amrit, P. Pinki, A. Kunal, J. & Sanjay, A. (2014). A Performance Analysis of MapReduce Task with Large Number of Files Dataset in Big Data Using Hadoop. In *Proceedings of the 4th International Conference on Communication Systems and Network Technologies*, (pp. 587-591).

Barbosa, T. M. S., Souza, R., Cruz, S. M. S., Campos, M. L., & Les Cottrell, R. (2015). Applying Data Warehousing and Big Data Techniques to Analyze Internet Performance. In *Proceedings of the 4th International Conference on Internet Applications, Protocols and Services*, (pp. 31-36).

Bacon, T. (2013). Big Bang? When 'Big Data' Gets Too Big. Available via: *http://www.eyefortravel.com/mobile-and-technology/big-bang-when-%E2%80%98big-data%E2%80%99-gets-too-big.*

Bloem, J., Van Doorn, M., Duivestein, S., & Van Ommeren, E. (2012). Creating Clarity with Big Data. *ACM Transactions on Computer Systems (TOCS)*, 26(2), 4.

Beyer, M. A., & Laney, D. (2012). The Importance of "Big Data": A Definition. *International Journal of Information Management*, *35*(2), 137-144.

Borkar, V., Carey, M. J., & Li, C. (2012, March). Inside Big Data Management: Ogres, Onions, or Parfaits. In *Proceedings of the 15th International Conference on Extending Database Technology* (pp. 3-14).

Biehn, N. (2013). The Missing Vs in Big Data: Viability and Value. *Science,* 343(6176), 1203-1205.

Chen, S. M., & Hwang, J. R. (2000). Temperature Prediction Using Fuzzy Time Series. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *30*(2), 263-275.

Dhanashri, V. (2015). Data Structure for Representation of Big Data of Weather Forecasting: A Review., *International Journal of Computer Science Trends and Technology (IJCST),* 3(6), 2347-8578.

DeWitt, D., & Gray, J. (1992). Parallel Database Systems: The Future of High Performance Database Systems. *Communications of the ACM*, *35*(6), 85-98.

Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Conference on Symposium on Operating Systems Design and Implementation* (pp. 10-100).

Dean, J. & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1), 107–113.

Elena, G., Florina, C., Anca, A., & Manole, V. (2012). Perspectives on Big Data and Big Data Analytics. *Database Systems Journal*, *3*(4), 3-14.

Einav, L., & Levin, J. (2014). The Data Revolution and Economic Analysis. *Innovation Policy and the Economy*, *14*(1), 1-24.

Falk, M. (2014). Impact of Weather Conditions on Tourism Demand in The Peak Summer Season Over the Last 50 Years. *Tourism Management Perspectives*, (9), 24-35.

Francisci, M., Lucchese, C. & Baraglia, R. (2010). Scaling Out All Pairs Similarity Search with MapReduce. *Large-Scale Distributed Systems for Information Retrieval*, 10(25).

Plimpton, S. J., & Devine, K. D. (2011). MapReduce in MPI for Large-Scale Graph Algorithms. *Parallel Computing*, *37*(9), 610-632.

Gupta, M., & George, J. F. (2016). Toward the Development of a Big Data Analytics Capability. *Information & Management*, *53*(8), 1049-1064.

Ghemawat, S., & Gobioff, H. (2003, October). The Google File System. *ACM SIGOPS Operating Systems Review*, 37(5), 29-43.

González-Bailón, S. (2013). Social Science in The Era of Big Data. *Policy & Internet*, 5(2), 147-160.

George, D. J. (1993). Weather and Mountain Activities. *Weather*, *48*(12), 404-410.

Genovese, Y., & Prentice, S. (2012). Pattern-Based Strategy: Getting Value from Big Data (G00214032), (p. 5).

Graham-Rowe, D., Goldston, D., Doctorow, C., Waldrop, M., Lynch, C., Frankel, F., & Rhee, S. Y. (2008). Big Data: Science in the Petabyte Era. *Nature*, *455*(7209), 8-9.

Grolinger, K., L'Heureux, A., Capretz, M. A., & Seewald, L. (2016). Energy Forecasting for Event Venues: Big Data and Prediction Accuracy. *Energy and Buildings*, (112), 222-233.

Hong, R. Y., Paunonen, S. V., & Slade, H. P. (2008). Big Five Personality Factors and the Prediction of Behavior: A Multitrait–Multimethod Approach. *Personality and Individual Differences*, *45*(2), 160-166.

Hassani, H., & Silva, E. S. (2015). Forecasting with Big Data: A Review. *Annals of Data Science*, *2*(1), 5-19.

Hu, H., Wen, Y., Chua, T. S., & Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. *IEEE Access*, (2), 652-687.

Hey, T., Tansley, S., & Tolle, K. M. (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery. In *Proceedings of the 3th International Symposium on Information Management in a Changing World* (pp. 1).

Hunter, P. (2013). Journey to the Centre of Big Data. *Engineering & Technology*, 8(3), 56-59.

Hassani, H., & Silva, E. S. (2015). Forecasting with Big Data: A Review. *Annals of Data Science*, *2*(1), 5-19.

Janssen, M., van der Voort, H., & Wahyudi, A. (2017). Factors Influencing Big Data Decision-Making Quality. *Journal of Business Research*, 70(C), 338-345.

Halim, Z., Baig, R., & Bashir, S. (2006, November). Sonification: A Novel Approach Towards Data Mining. In Emerging Technologies. In *Proceedings of 19th International Conference on* (pp. 548-553).

Infosys, L. (2013). Big data: Challenges and Opportunities. *Proceedings of the VLDB Endowment*, 5(12), 2032-2033.

James, N., Yanxing, H., Jane, J. & Pak, W. (2014). Deep Neural Network Based Feature Representation for Weather Forecasting. In *Proceedings of the International Conference on Artificial Intelligence* (p. 1).

Jiang, H., Chen, Y., Qiao, Z., Weng, T. H., & Li, K. C. (2015). Scaling up MapReduce-Based Big Data Processing on Multi-GPU Systems. *Cluster Computing*, 18(1), 369-383.

Jain, A., & Subbulakshmi, T. (2016). Analysis and Review the Data Using Big Data Hadoop. *International Journal of Database Theory and Application*, *9*(5), 203-212.

Jararweh, Y., Alsmadi, I., Al-Ayyoub, M., & Jenerette, D. (2014, November). The Analysis of Large-Scale Climate Data: Jordan Case Study. In *Proceedings of the 11th International Conference on* (pp. 288-294).

Kwong, K., Wong, M., Liu, J. & Chan, P. (2012). An Artificial Neural Network with Chaotic Oscillator for Wind Shear Alerting. *Journal of Atmospheric and Oceanic Technology*, 29(10), 1518–1531.

Kaisler, S., Armour, F., Espinosa, J. A. & Money, W. (2013, January). Big Data: Issues and Challenges Moving Forward. In *Proceedings of the 46th Hawaii International Conference on System Sciences* (pp. 995-1004).

Kearney, A. T. (2013). Big Data and the Creative Destruction of Today's Business Models. *The Journal of Strategic Information Systems*, 24(3), 149-157.

Kulendran, N. & Wong, K. (2005). Modeling Seasonality in Tourism Forecasting," *Journal of Travel Research*, vol. 44(2), 163-170.

Kobielus, J. G. (2012). The Forrester Wave: Enterprise Hadoop Solutions, Q1 2012. *Forrester Research*.

Mouthaan, N. (2012). Effects of Big Data Analytics on Organizations' Value Creation. *Academy of management journal*, 34(3), 555-590.

Knulst, J. (2012). De Stand van Hadoop. In *Proceedings of the International Conference on Computational Intelligence and Computing Research*, (pp. 1-4).

Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Ching, C., Choi, A., & Joshi, I. (2015, January). Impala: A Modern, Open-Source SQL Engine for Hadoop. In *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research* (pp. 1-9)

Katal, A., Wazid, M., & Goudar, R. H. (2013, August). Big Data: Issues, Challenges, Tools and Good Practices. In *Contemporary Computing (IC3).* In *Proceedings of the 16th International Conference on Contemporary Computing* (pp. 404-409).

Lee, K., Hong, B., Lee, J., & Jang, Y. (2015, August). A Floating Population Prediction Model in Travel Spots Using Weather Big Data. In *Proceedings of the 15th International Conference on* Big *Data and Cloud Computing (BDCloud)* (pp. 118-124).

Li, K., Grant, C., Wang, D. Z., Khatri, S., & Chitouras, G. (2013, June). Gptext: Greenplum Parallel Statistical Text Analysis Framework. In *Proceedings of the 2nd Workshop on Data Analytics in the Cloud* (pp. 31-35).

Marjit, U., Sharma, K., & Mandal, P. (2015). Data Transfers in Hadoop: A Comparative Study. *Open Journal of Big Data (OJBD)*, *1*(2), 34-46.

Marr, B. (2015). Big Data: Using Smart Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance. *John Wiley & Sons.*

Madden, S. (2012). From Databases to Big Data. *IEEE Internet Computing*, *16*(3), 4-6.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big Data. *The Management Revolution. Harvard Bus Rev*, *90*(10), 61-67.

McKinsey Global Institute (2012). Big Data: The Next Frontier for Innovation, Competition, and Productivity. *The McKinsey Global Institute.*

MacAlpine, H. K., Gordân, R., Powell, S. K., Hartemink, A. J., & MacAlpine, D. M. (2010). Drosophila ORC Localizes to Open Chromatin and Marks Sites of Cohesin Complex Loading. *Genome research*, *20*(2), 201-211.

Matsuoka, S., Sato, H., Tatebe, O., Koibuchi, M., Fujiwara, I., Suzuki, S., & Ueno, K. (2014). Extreme Big Data (EBD): Next Generation Big Data Infrastructure

Technologies Towards Yottabyte. *Supercomputing Frontiers and Innovations*, *1*(2), 89-107.

Mandal, B., Sahoo, R. K., & Sethi, S. (2017). Scalable Big Data Analysis in Cloud Environment: A Review. *IJRCCT*, *5*(12), 623-630.

Netezza, N. and Marlborough, M. (2013). MA, USA [Online]. Available: *http://www-01.ibm.com/software/data/netezza.2013*.

Kim, G. H., Trimi, S., & Chung, J. H. (2014). Big-Data Applications in The Government Sector. *Communications of the ACM*, *57*(3), 78-85.

Olson, S., & Riordan, D. G. (2012). Engage to Excel: Producing One Million Additional College Graduates with Degrees in Science, Technology, Engineering, and Mathematics. Report to the President. *Executive Office of the President*.

Rose, S. A., Poynter, P. S., Anderson, J. W., Noar, S. M., & Conigliaro, J. (2013). Physician Weight Loss Advice and Patient Weight Loss Behavior Change: A Literature Review and Meta-Analysis of Survey Data. *International Journal of Obesity*, *37*(1), 118-128.

Ferraro, R., Sato, T., Brasseur, G., Deluca, C., & Guilyardi, E. (2003). Modeling the Earth System. *Computing in Science & Engineering*, *6*(1), 18-28.

Richards, N. M., & King, J. H. (2014). Big Data Ethics. *Wake Forest L. Rev.*, *49*, 393.

Radhika, Y. & Shashi, D. (2013). Atmospheric Temperature Prediction Using Support Vector Machines. *International Journal of Computer Theory and Engineering*, 1(1), 55.

Robert C., Hairong K., Sanjay R., Konstantin S. & Suresh S. (2010). The Hadoop Distributed File System. In *Proceedings of the 26th Symposium on Mass Storage Systems and Technologies*, (pp. 1-10).

Schmarzo, B. (2013). Big Data: Understanding How Data Powers Big Business. *John Wiley & Sons.*

Slava R.,Drew S., Z., Yousaf K., Christoph B., & Earo, W. (2014). Forecast: Forecasting Functions for Time Series and Linear Models. *R package version*, *5.*

Sicular, S. (2013). Gartner's Big Data Definition Consists of Three Parts, Not to Be Confused with Three" V" s. *Gartner, Inc*, *27*.

Samuel, G. (2014). Weathering A New Era of Big Data. *Communications of the ACM*, *57*(9), 12-14.

Shafer, J., Rixner, S., & Cox, A. L. (2010, March). The Hadoop Distributed Filesystem: Balancing Portability and Performance. In *Performance of the International Symposium on Analysis of Systems & Software (ISPASS)*, (pp. 122-133).

Sagiroglu, S., & Sinanc, D. (2013, May). Big Data: A review. In *Performance of the International Conference on Collaboration Technologies and Systems* (pp. 42-47).

Tucker, P. (2013). The Future is Not a Destination. *The Futurist Magazine's top*, *10*.

Tang, X., Wang, L., & Geng, Z. (2015). A Reduce Task Scheduler for MapReduce with Minimum Transmission Cost Based on Sampling Evaluation. *International Journal of Database Theory and Application*, 8(1), 1-10.

Tom, W. (2012). Hadoop: The Definitive Guide. " O'Reilly Media, Inc.".

Villars, R. L., Olofson, C. W., & Eastwood, M. (2011). Big Data: What it is and Why You Should Care. *White Paper, IDC*, 14.

Wang, L., Tao, J., Ranjan, R., Marten, H., Streit, A., Chen, J., & Chen, D. (2013). G-Hadoop: MapReduce Across Distributed Data Centers for Data-Intensive Computing. *Future Generation Computer Systems*, *29*(3), 739-750.

White, B., Yeh, T., Lin, J., & Davis, L. (2010, July). Web-Scale Computer Vision Using MapReduce for Multimedia Data Mining. In *Proceedings of the 10th International Workshop on Multimedia Data Mining* (p. 9).

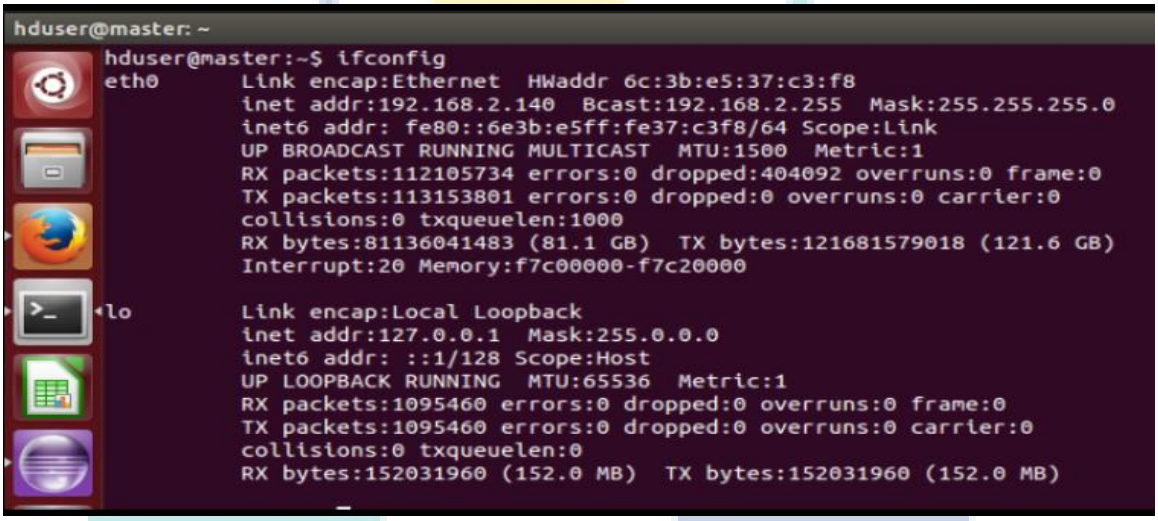
Yoon, J. H., & Kim, S. R. (2011, September). Improved Sampling for Triangle Counting with MapReduce. In *Proceedings of the International Conference on Hybrid Information Technology* (pp. 685-689).

APPENDIX A
Install Multi Nodes Hadoop Cluster on Ubuntu

This explains the setup of the Hadoop Multi-Node cluster on a distributed environment. We are explaining the Hadoop cluster environment using three computers one NameNode (master) and two DataNodes (slaves). given below are their IP addresses.

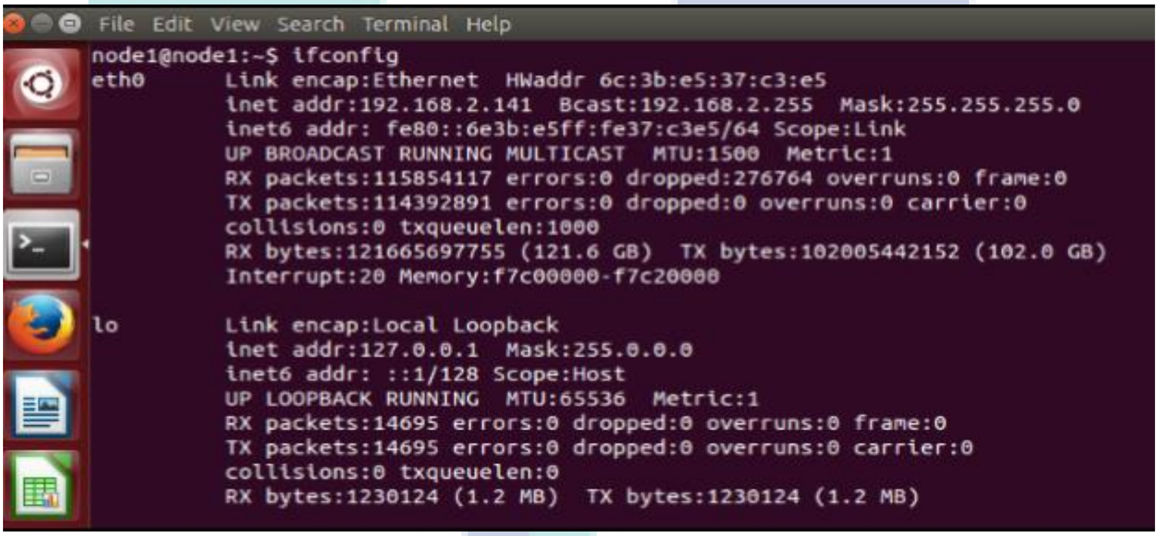
Network

Master ip address



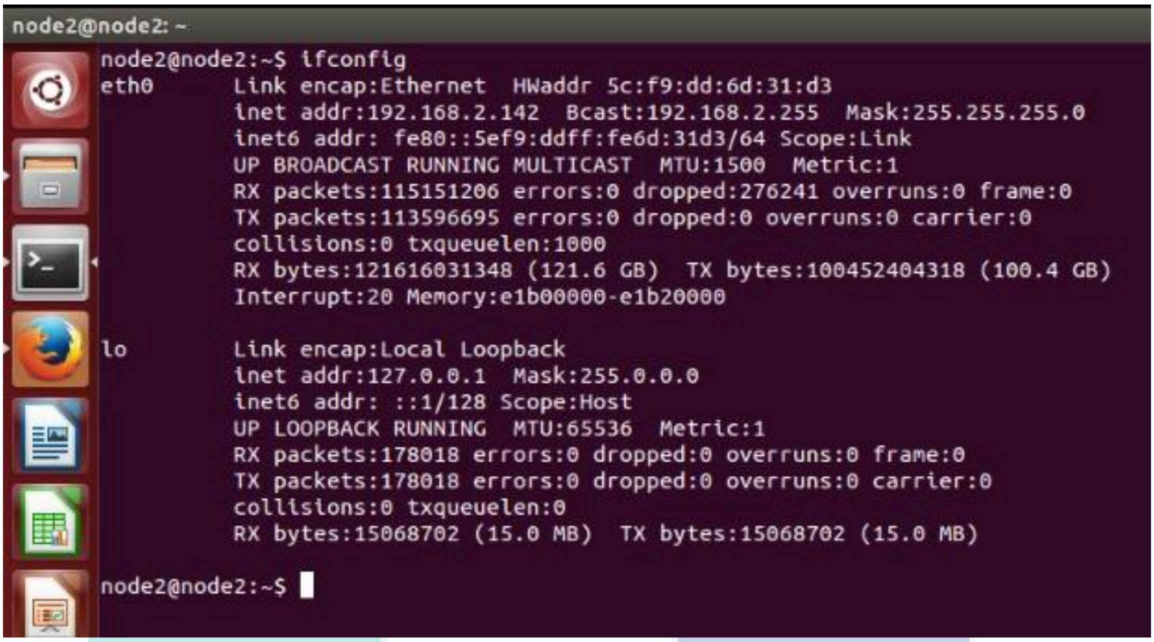
```
hduser@master: ~
hduser@master:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 6c:3b:e5:37:c3:f8
          inet addr:192.168.2.140  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6e3b:e5ff:fe37:c3f8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:112105734 errors:0 dropped:404092 overruns:0 frame:0
          TX packets:113153801 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:81136041483 (81.1 GB)  TX bytes:121681579018 (121.6 GB)
          Interrupt:20 Memory:f7c00000-f7c20000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1095460 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1095460 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:152031960 (152.0 MB)  TX bytes:152031960 (152.0 MB)
```

Slave1 ip address



```
File  Edit  View  Search  Terminal  Help
node1@node1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 6c:3b:e5:37:c3:e5
          inet addr:192.168.2.141  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6e3b:e5ff:fe37:c3e5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115854117 errors:0 dropped:276764 overruns:0 frame:0
          TX packets:114392891 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:121665697755 (121.6 GB)  TX bytes:102005442152 (102.0 GB)
          Interrupt:20 Memory:f7c00000-f7c20000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:14695 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14695 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1230124 (1.2 MB)  TX bytes:1230124 (1.2 MB)
```

Slave2 ip address



- **INSTALLATION PREREQUISITES**

Installing Java on Master and Slaves

$ sudo add-apt-repository ppa:webupd8team/java

$ sudo apt-get update

$ sudo apt-get install oracle-java7-installer

Updata Java runtime

$ sudo update-java-alternatives -s java-7-oracle

- **SETTING UP A HADOOP USER**

Hadoop talks to other nodes in the cluster using no-password ssh. By having Hadoop run under a specific user context, it will be easy to distribute the ssh keys around in the Hadoop cluster. Create a user hduser on master as well as slave nodes.

Create hadoopgroup

$ sudo addgroup hadoopgroup

Create hduser user

$ sudo adduser —ingroup hadoopgroup hduser

- **SSH**

The next step will be to generate a ssh key for password-less login between master and slave nodes. The researcher Run the following commands only **on master** node. Run the last two commands for **each slave node**. Password less ssh should be working before proceed with further steps.

Login as hdpuser

$ su – hduser

#Generate a ssh key for the user

$ ssh-keygen -t rsa -P ""

#Authorize the key to enable password less ssh

$ cat /home/hduser/.ssh/id_rsa.pub >> /home/hduser/.ssh/authorized_keys

$ chmod 600 authorized_keys

#Copy this key to slave1 and slave2 to enable password less ssh

$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave1

$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave2

#Test the password less ssh using following command.

$ ssh slave1

$ ssh slave2

- **Download and Install Hadoop binaries on Master and Slave nodes**

Researcher pick the best mirror site to download the binaries from Apache Hadoop, and download the stable/hadoop-2.7.1.tar.gz for installation. Researcher did this step on master and every slaves node. It can download the file once and the distribute it in each slaves' node using scp command.

$ cd /home/hduser

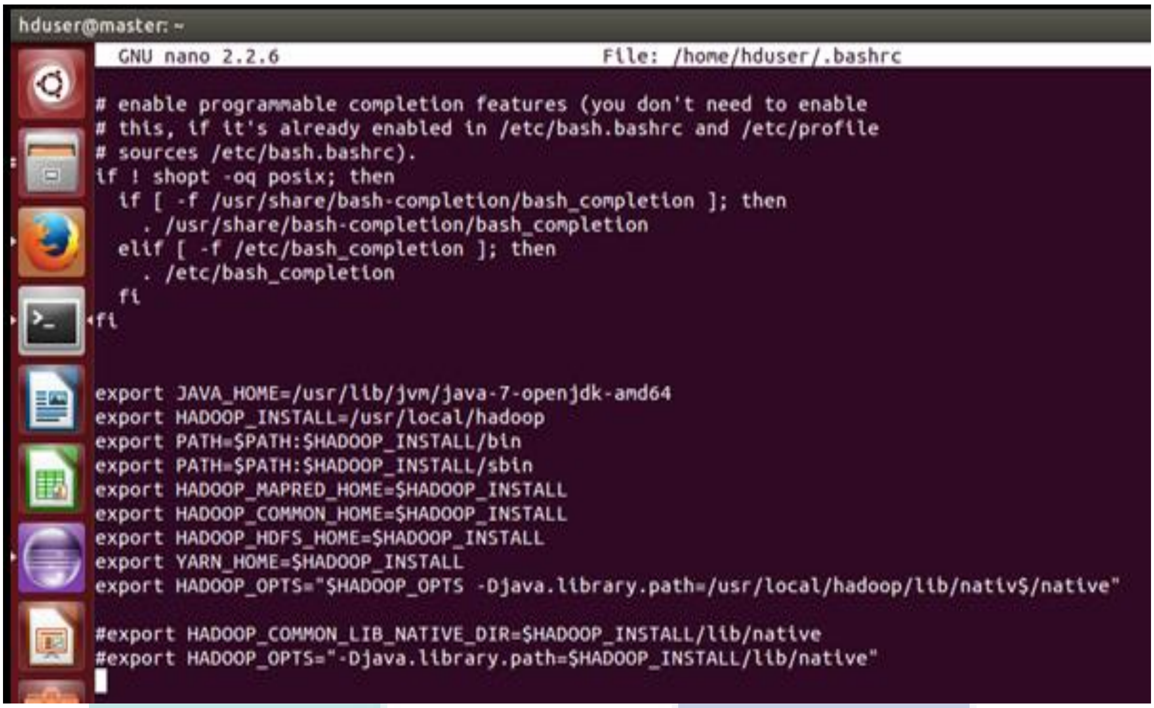
$sudo wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.7.1/hadoop-2.7.1.tar.gz

$ tar xvf hadoop-2.7.1.tar.gz

$ mv hadoop-2.7.1 hadoop

- **Researcher Setup Hadoop Environment on Master and Slaves Node**

Researcher copy and paste following lines into the. bashrc file under /home/hduser. Researcher did this step on master and every slaves node.



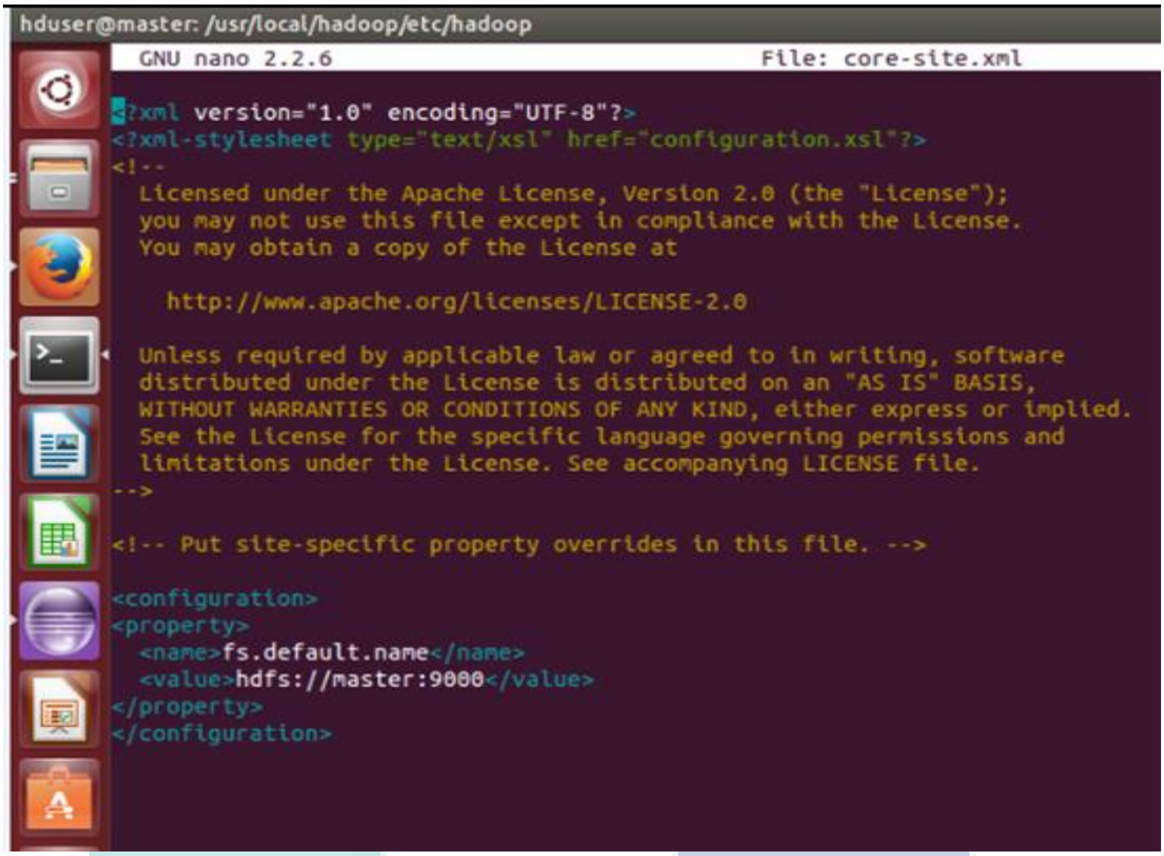
Researcher updated hadoop-env.sh on Master and Slave Nodes

Update JAVA_HOME in /home/hduser/hadoop/etc/hadoop/hadoop_env.sh to following. Researcher did this step on master and every slaves node.

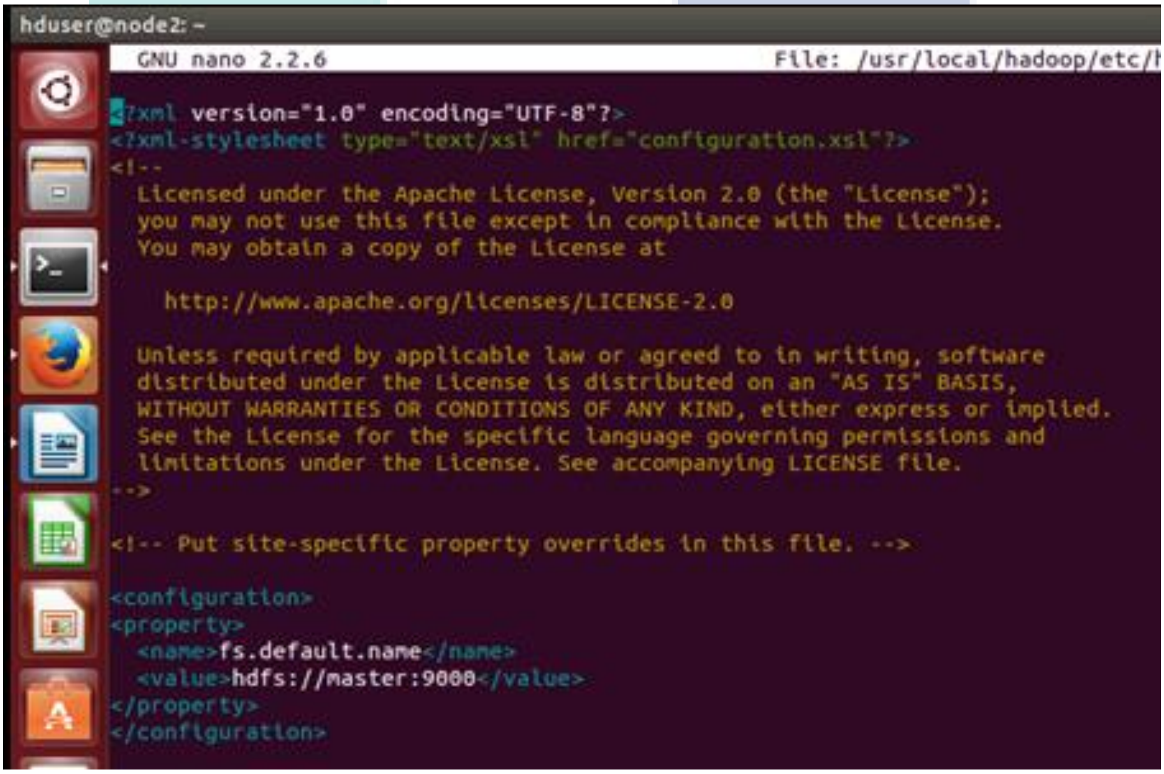
export JAVA_HOME=/usr/lib/jvm/java-7-oracle

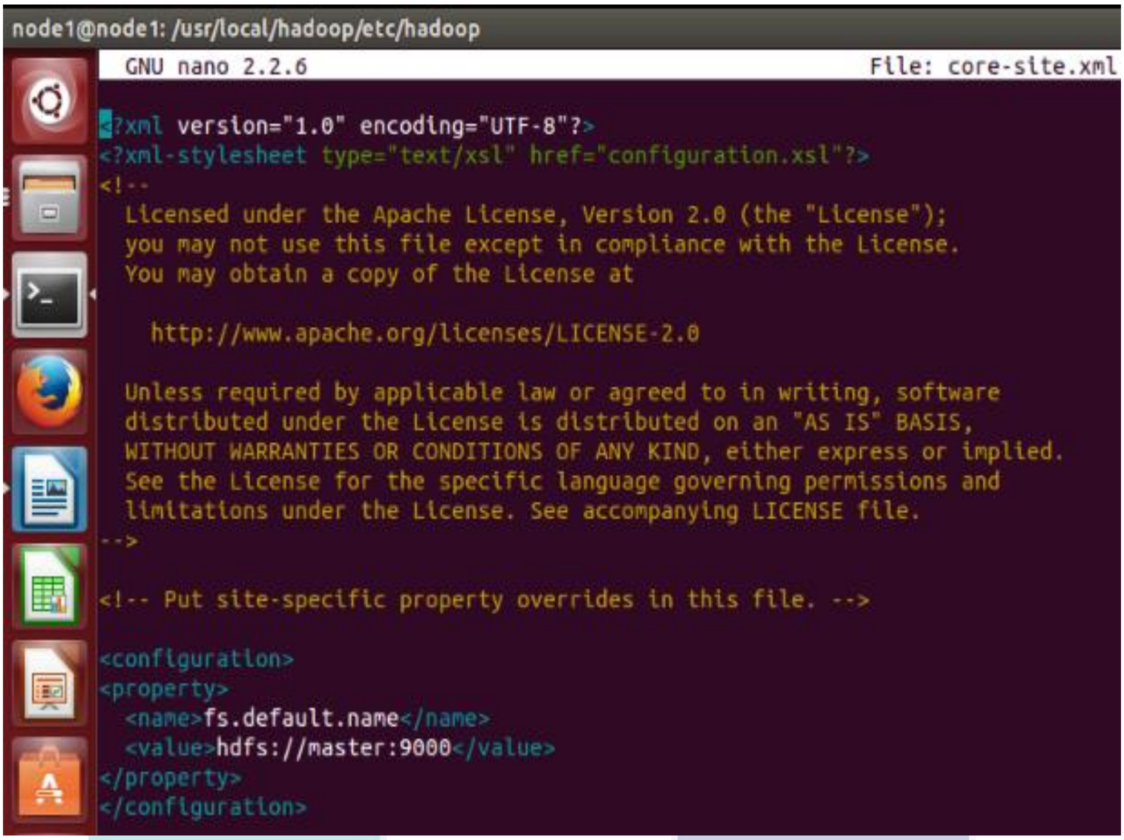
- **Researcher update core-site.xml on Master and Slave nodes with following options. Master and slave nodes should all be using the same value for this property fs.defaultFS, and should be pointing to master node only.**
- **On master**

```
hduser@master: /usr/local/hadoop/etc/hadoop
  GNU nano 2.2.6                                File: hadoop-env.sh

# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done

# The maximum amount of heap to use, in MB. Default is 1000.
#export HADOOP_HEAPSIZE=
#export HADOOP_NAMENODE_INIT_HEAPSIZE=""

# Extra Java runtime options.  Empty by default.
export HADOOP_OPTS="$HADOOP_OPTS -Djava.net.preferIPv4Stack=true"
```

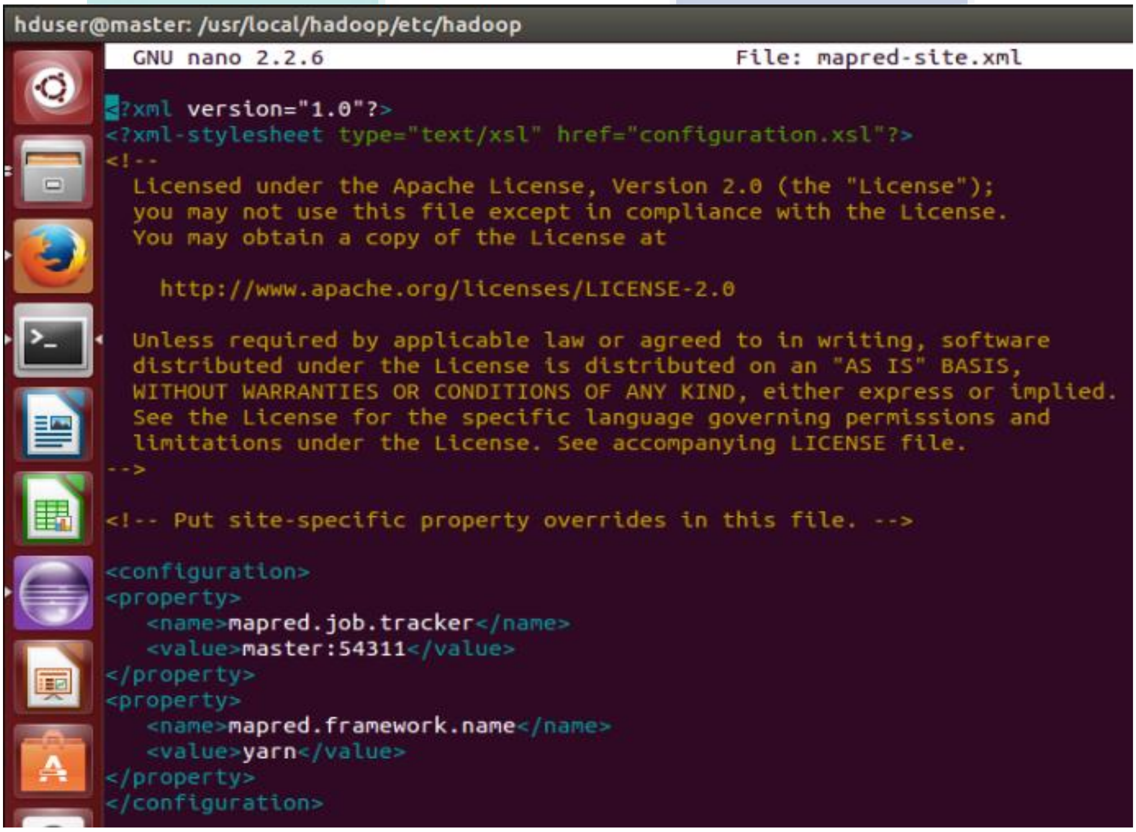

hduser@master: /usr/local/hadoop/etc/hadoop
GNU nano 2.2.6
File: core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
</configuration>

- **On slaves**


hduser@node2: ~
GNU nano 2.2.6
File: /usr/local/hadoop/etc/
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
</configuration>



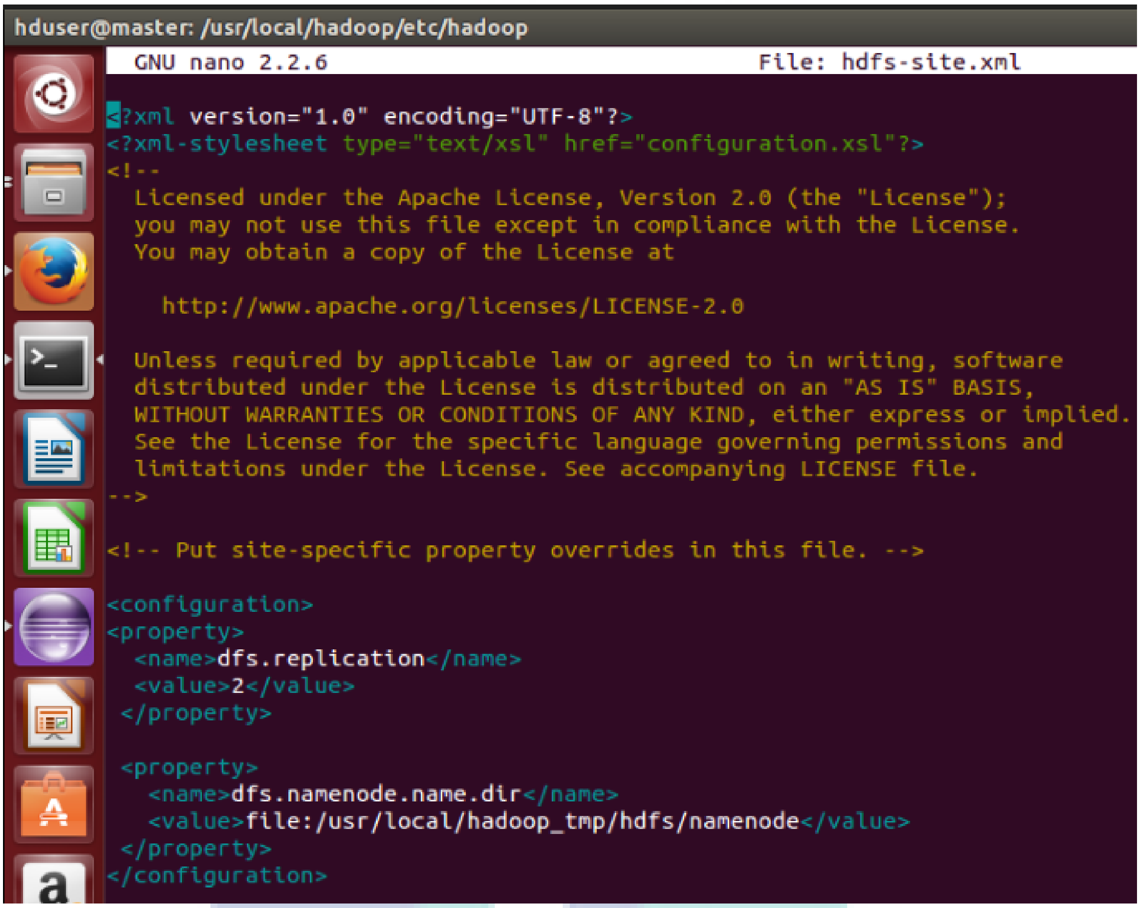
- **Researcher update mapred-site.xml on Master node only with following options.**



- **Add/update hdfs-site.xml on Master and Slave Nodes. We will be adding following three entries to the file.**

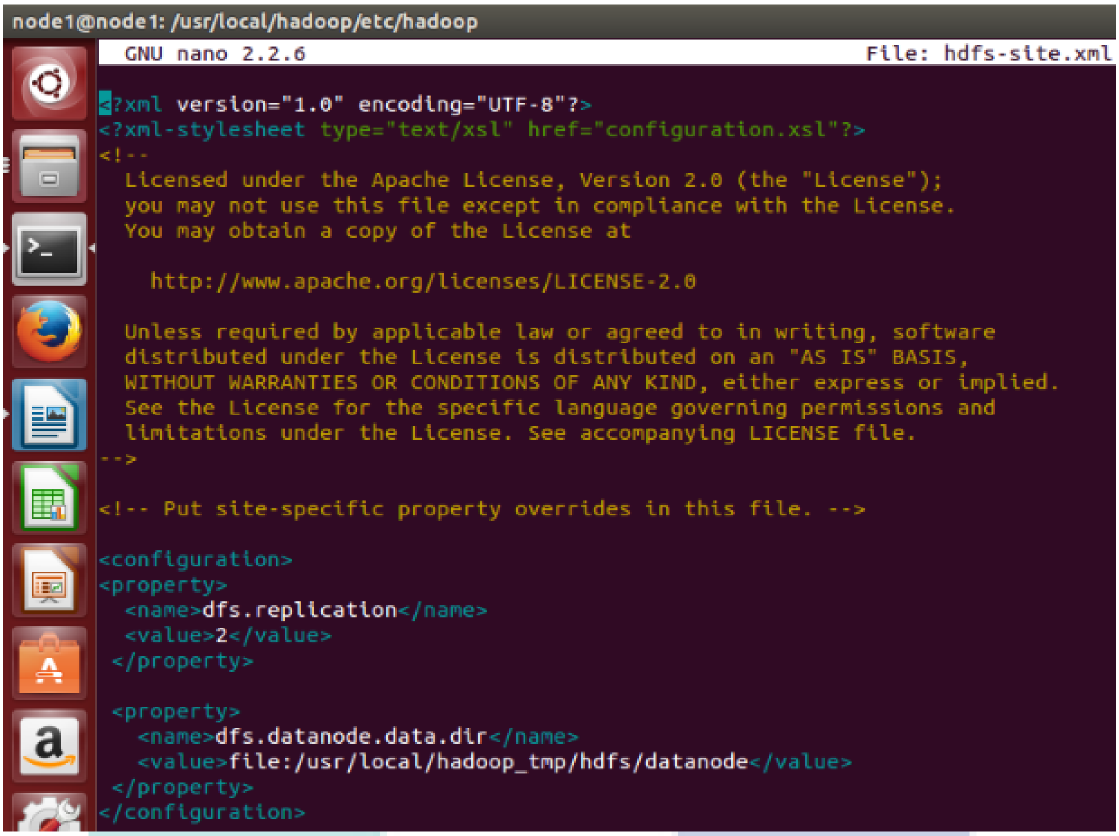
researcher using a replication factor of 2. That means for every file stored in HDFS, there will be one redundant replication of that file on some other node in the cluster.

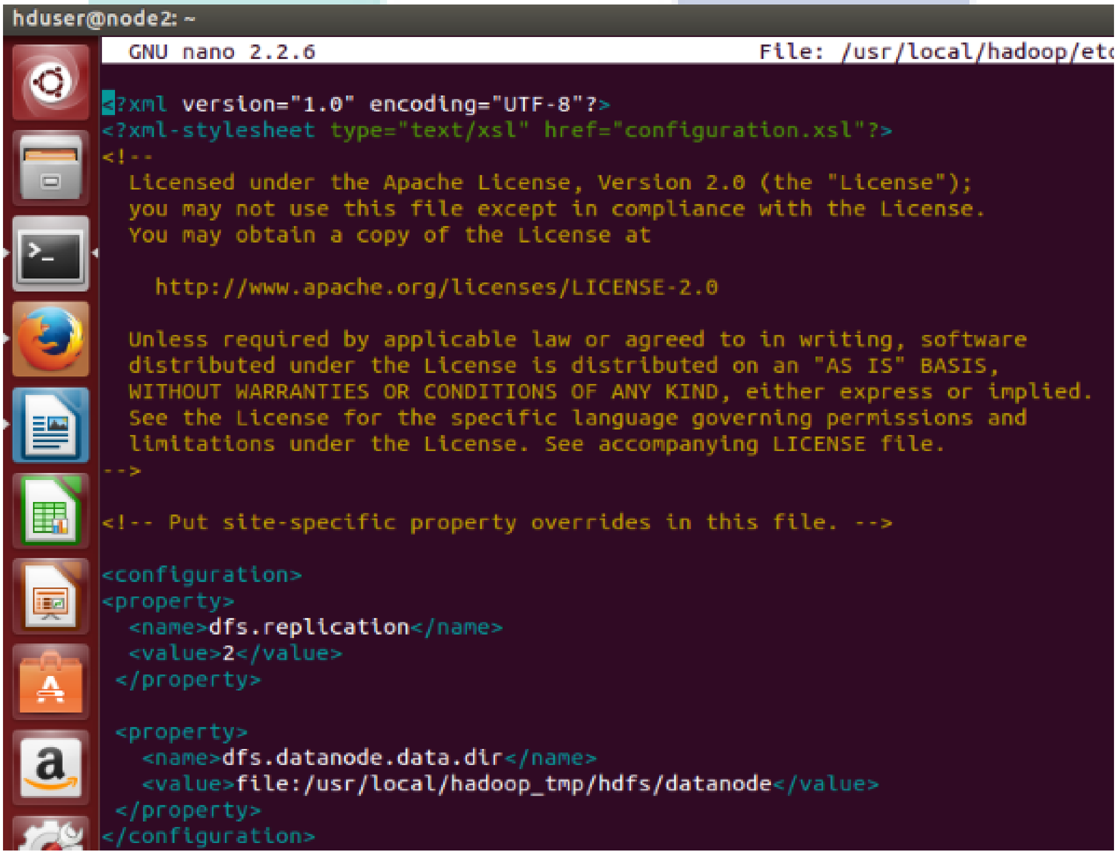
On master



```
hduser@master: /usr/local/hadoop/etc/hadoop
  GNU nano 2.2.6                             File: hdfs-site.xml

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.replication</name>
  <value>2</value>
 </property>

 <property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
 </property>
</configuration>
```

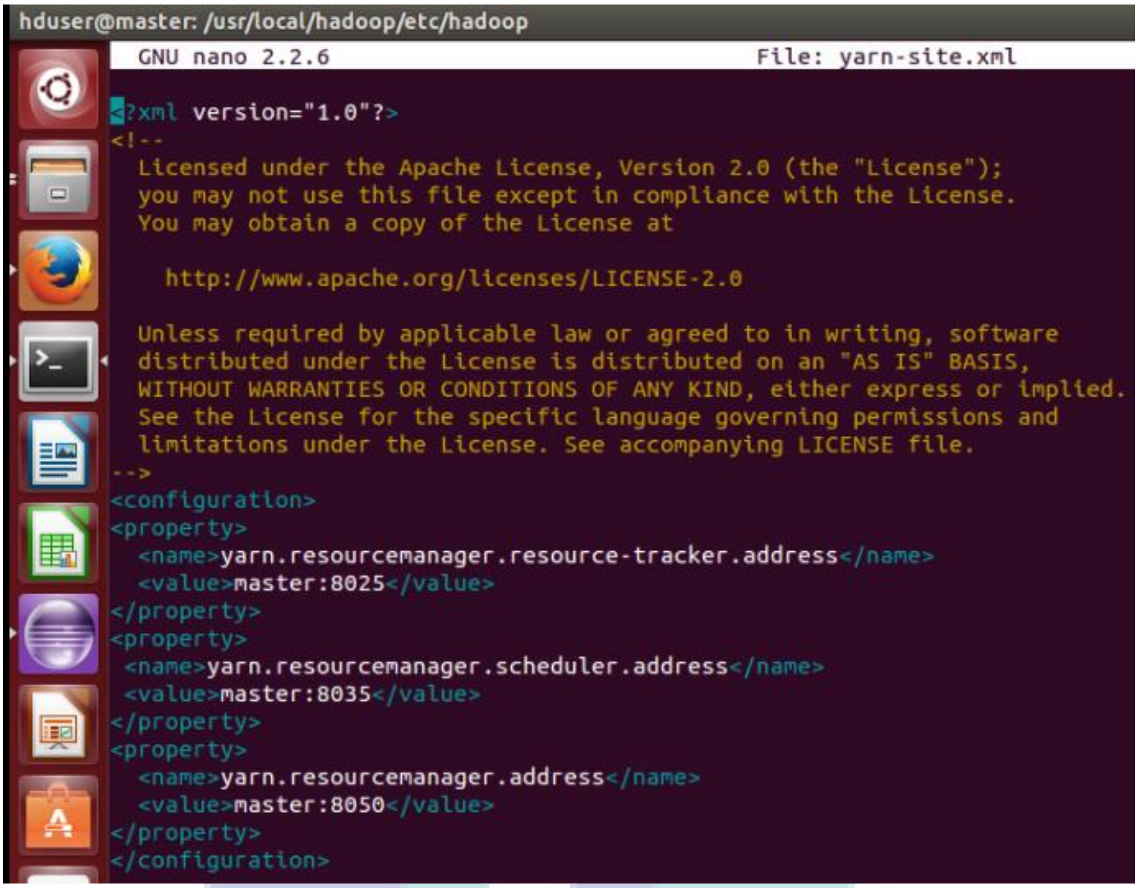
Slaves


node1@node1: /usr/local/hadoop/etc/hadoop
GNU nano 2.2.6
File: hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>

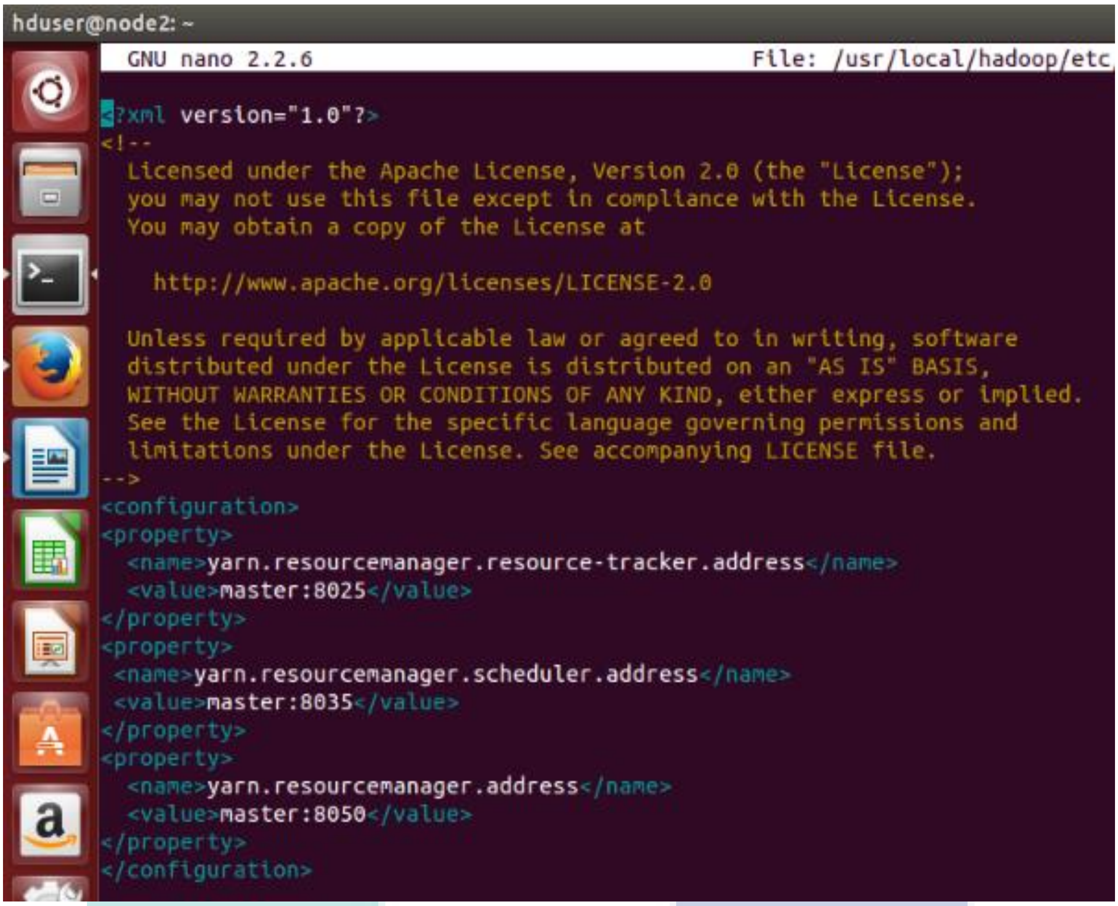

hduser@node2: ~
GNU nano 2.2.6
File: /usr/local/hadoop/etc
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>

- **Researcher add yarn-site.xml on Master and Slave Nodes. This file is required for a Node to work as a Yarn Node. Master and slave nodes should all be using the same value for the following properties, and should be pointing to master node only.**

On master



On slaves



- **Format the Namenode**

  Before starting the cluster, researcher format the Namenode. Use the following command only on master node:

  $ hdfs namenode -format

- **Start the Distributed Format System**

  Researcher run the following on master node command to start the DFS.

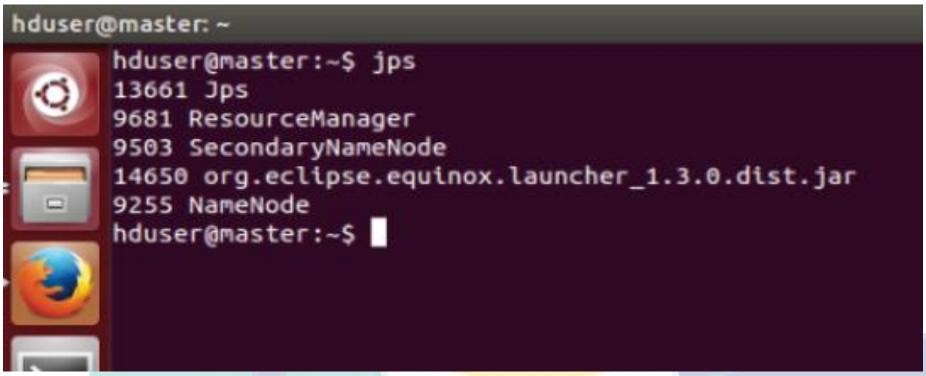
  $ start-all.sh

Researcher observes the output to ascertain that it tries to start datanodes on slave nodes one by one. To validate the success, run following command on master nodes, and slave node.

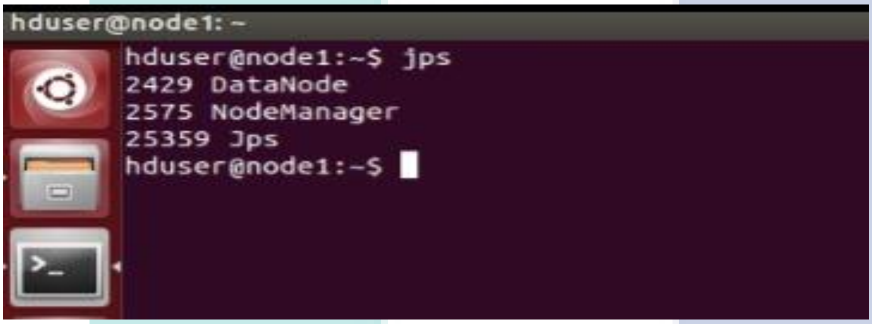
$ su – hduser

$ jps

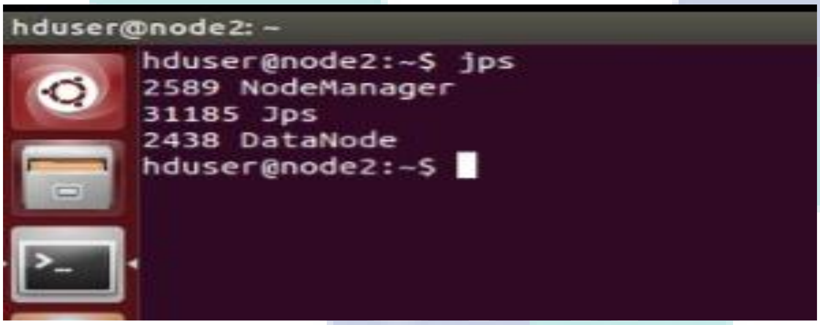
Master



Slave1

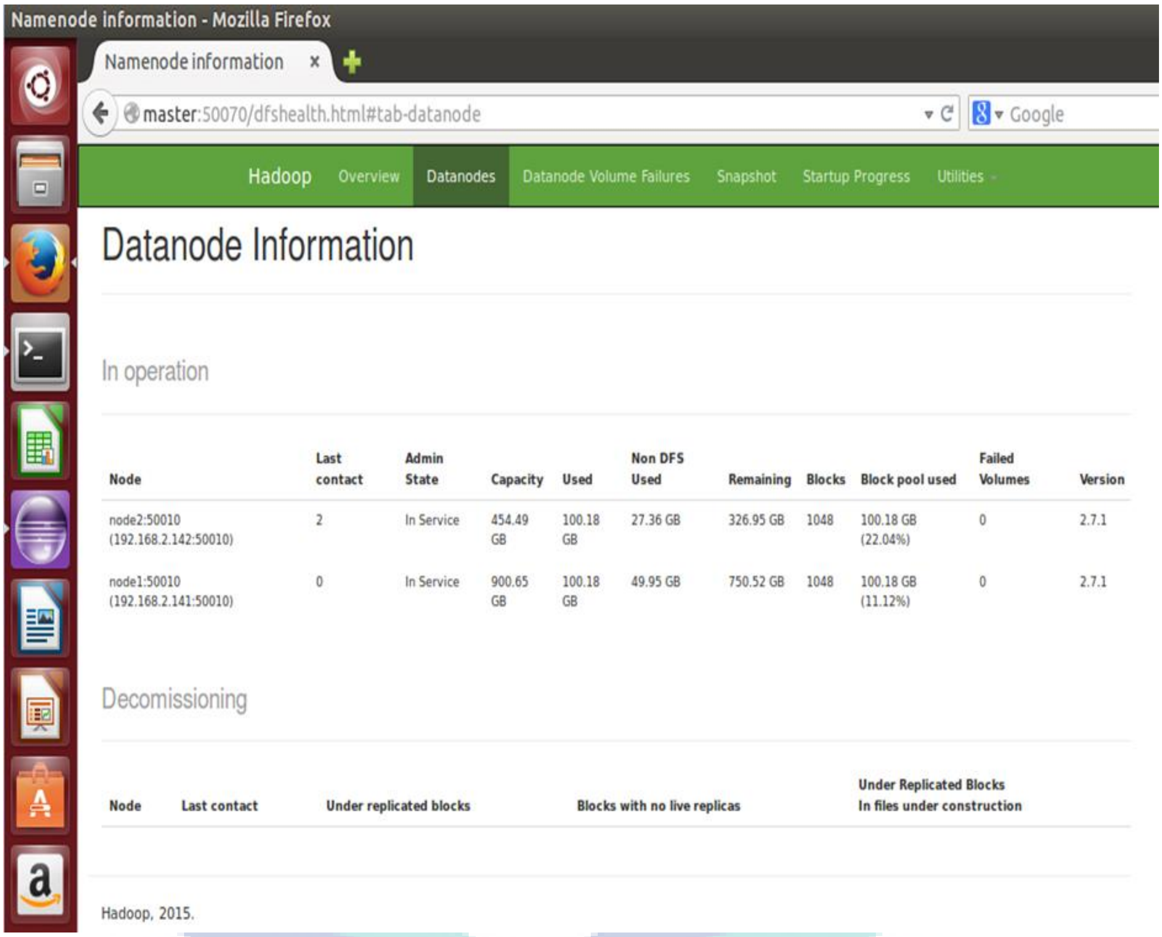


Slave2

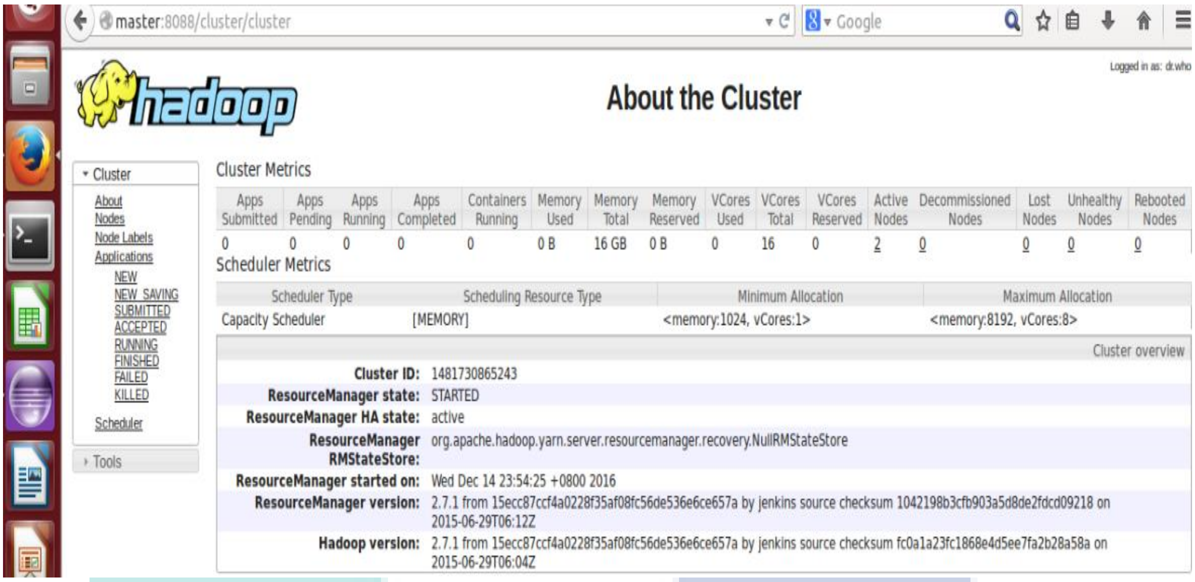


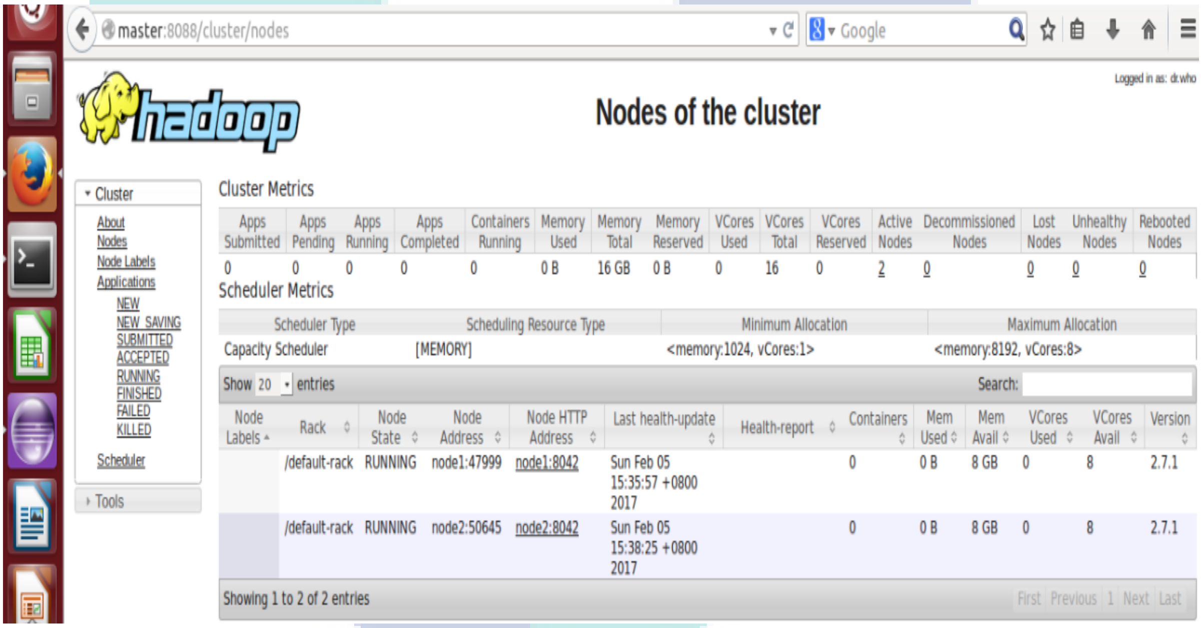
- **Accessing Hadoop on Browser**

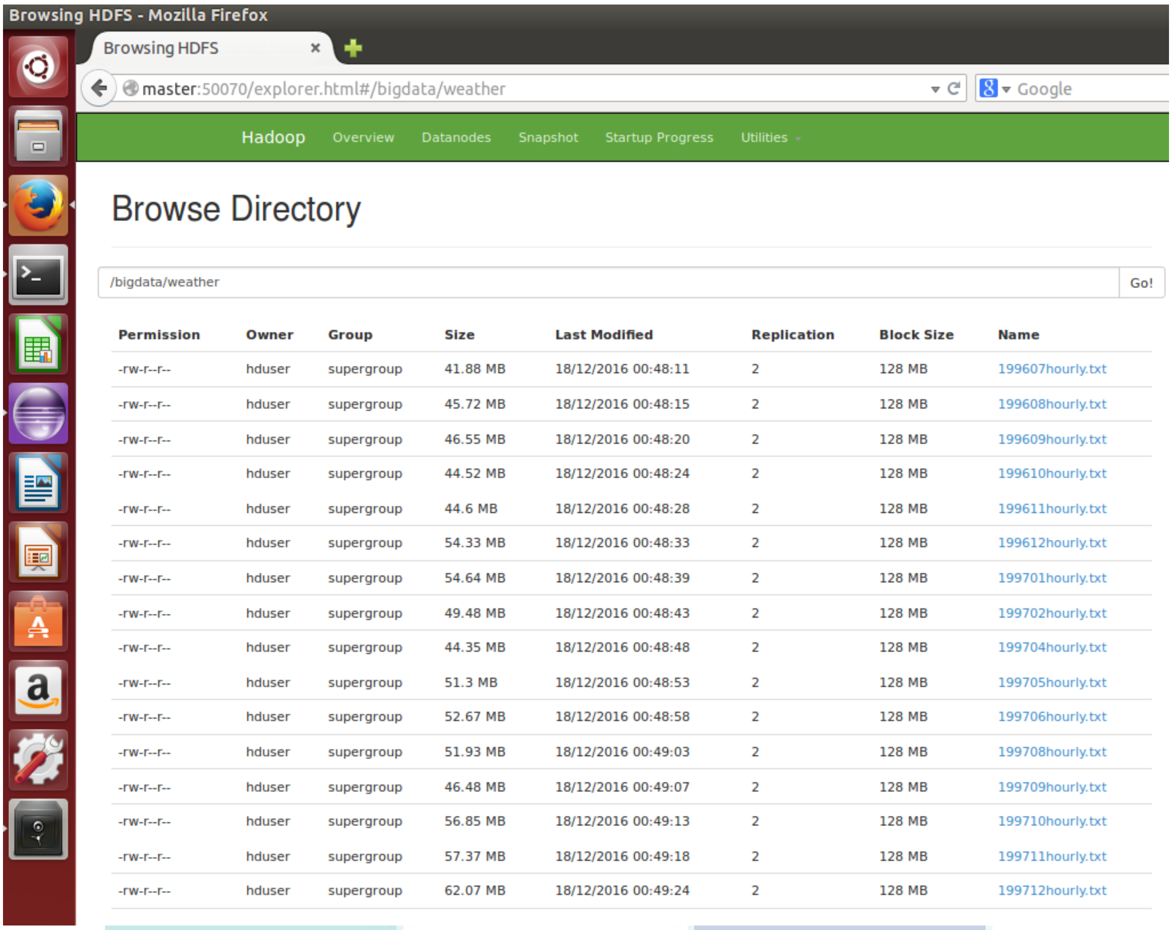
The default port number to access Hadoop is 50070. Use the following URL http://master:50070/cluster/nodes to get Hadoop services on your browser.



http://master:8088/cluster/nodes







Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hduser	supergroup	41.88 MB	18/12/2016 00:48:11	2	128 MB	199607hourly.txt
-rw-r--r--	hduser	supergroup	45.72 MB	18/12/2016 00:48:15	2	128 MB	199608hourly.txt
-rw-r--r--	hduser	supergroup	46.55 MB	18/12/2016 00:48:20	2	128 MB	199609hourly.txt
-rw-r--r--	hduser	supergroup	44.52 MB	18/12/2016 00:48:24	2	128 MB	199610hourly.txt
-rw-r--r--	hduser	supergroup	44.6 MB	18/12/2016 00:48:28	2	128 MB	199611hourly.txt
-rw-r--r--	hduser	supergroup	54.33 MB	18/12/2016 00:48:33	2	128 MB	199612hourly.txt
-rw-r--r--	hduser	supergroup	54.64 MB	18/12/2016 00:48:39	2	128 MB	199701hourly.txt
-rw-r--r--	hduser	supergroup	49.48 MB	18/12/2016 00:48:43	2	128 MB	199702hourly.txt
-rw-r--r--	hduser	supergroup	44.35 MB	18/12/2016 00:48:48	2	128 MB	199704hourly.txt
-rw-r--r--	hduser	supergroup	51.3 MB	18/12/2016 00:48:53	2	128 MB	199705hourly.txt
-rw-r--r--	hduser	supergroup	52.67 MB	18/12/2016 00:48:58	2	128 MB	199706hourly.txt
-rw-r--r--	hduser	supergroup	51.93 MB	18/12/2016 00:49:03	2	128 MB	199708hourly.txt
-rw-r--r--	hduser	supergroup	46.48 MB	18/12/2016 00:49:07	2	128 MB	199709hourly.txt
-rw-r--r--	hduser	supergroup	56.85 MB	18/12/2016 00:49:13	2	128 MB	199710hourly.txt
-rw-r--r--	hduser	supergroup	57.37 MB	18/12/2016 00:49:18	2	128 MB	199711hourly.txt
-rw-r--r--	hduser	supergroup	62.07 MB	18/12/2016 00:49:24	2	128 MB	199712hourly.txt

APPENDIX B

CODE

```
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.commons.lang.StringUtils;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
import java.util.Iterator;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;


public class AvgWeather {
public static void main(String[] args) throws IOException {
      JobConf conf = new JobConf(AvgWeather.class);
      conf.setJobName("Avg");

      conf.setOutputKeyClass(Text.class);
      conf.setMapOutputValueClass(IntWritable.class);

      conf.setMapperClass(AvgWeatherMapper.class);
      conf.setReducerClass(AvgWeatherReducer.class);

      FileInputFormat.addInputPath(conf, new Path(args[0]));
      FileOutputFormat.setOutputPath(conf, new Path(args[1]));

      JobClient.runJob(conf);
   }
}
public class AvgWeatherMapper extends MapReduceBase
      implements Mapper<LongWritable, Text, Text, IntWritable> {
public void map(LongWritable key, Text value,
      OutputCollector<Text, IntWritable> output, Reporter reporter)
      throws IOException {
      String[] line = value.toString().split(","); //split line by
comma into array
      String datepart = line[1]; //extract year
      String temp = line[10]; //extract temperature
      //String temp2 = line[13]; // Relative Humidity
      //String temp2 = line[8]; //extract Visibility

      if (StringUtils.isNumeric(temp2)) {
      output.collect(new Text(datepart), new
      IntWritable(Integer.parseInt(temp)));
      }

      /*if (StringUtils.isNumeric(Humid)) {
```

```
//output.collect(new Text(datepart), new
IntWritable(Integer.parseInt(Humid)));
      }*/

/*if (StringUtils.isNumeric(Visib)) {
//output.collect(new Text(datepart), new
IntWritable(Integer.parseInt(Visib)));
      }*/
}

   }

public class AvgWeatherReducer extends MapReduceBase
 implements Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterator<IntWritable>values,
OutputCollector<Text, IntWritable> output, Reporter reporter)
      throws IOException {
      int sumTemps = 0; //sum of all temps per key
      //int sumHumid = 0; //sum of all Humids per key
      //int sumVisib = 0; //sum of all Visibs per key
      int numItems = 0; //
      while (values.hasNext()) {
      sumTemps += values.next().get();
      numItems += 1;
      output.collect(key, new IntWritable(sumTemps / numItems));
      output.collect(key, new IntWritable(sumTemps));

      //output.collect(key, new IntWritable(sumHumid / numItems));
      //output.collect(key, new IntWritable(sumHumid));

      //output.collect(key, new IntWritable(sumVisib / numItems));
      //output.collect(key, new IntWritable(sumVisib));
}

      }
```

APPENDIX C

The Unstructured Dataset

Wban Number, YearMonthDay, Time, Station Type, Maintenance Indicator, Sky Conditions, Visibility, Weather Type, Dry Bulb Temp, Dew Point Temp, Wet Bulb Temp, % Relative Humidity, Wind Speed (kt), Wind Direction, Wind Char. Gusts (kt), Val for Wind Char., Station Pressure, Pressure Tendency, Sea Level Pressure, Record Type, Precip. Total

03013,19960701,0053,AO20,-,CLR ,10SM ,-,64,60.1,35, 87 , 7 ,180,-,0 ,26.30,-,162,AA,-

03013,19960701,0153,AO20,-,CLR ,10SM ,-,64.9,60.1,35, 84 , 10 ,190,-,0 ,26.30,6,153,AA,-

03013,19960701,0253,AO20,-,CLR ,10SM ,-,62.1,60.1,34.9, 93 , 8 ,200,-,0 ,26.29,-,150,AA,-

03013,19960701,0353,AO20,-,CLR ,10SM ,-,60.1,59,34.7, 96 , 3 ,310,-,0 ,26.29,-,151,AA,-

03013,19960701,0453,AO20,-,CLR ,10SM ,-,59,57.9,34.6, 96 , 0 ,000,-,0 ,26.30,5,154,AA,-

03013,19960701,0553,AO20,-,CLR ,10SM ,-,64,61,35, 90 , 0 ,000,-,0 ,26.30,-,155,AA,-

03013,19960701,0653,AO20,-,CLR ,10SM ,-,66.9,62.1,35.2, 84 , 6 ,310,-,0 ,26.31,-,162,AA,-

03013,19960701,0753,AO20,-,CLR ,10SM ,-,72,63,35.4, 73 , 5 ,310,-,0 ,26.31,3,160,AA,-

03013,19960701,0853,AO20,-,CLR ,10SM ,-,75.9,63,35.5, 64 , 6 ,270,-,0 ,26.31,-,156,AA,-

03013,19960701,0953,AO20,-,CLR ,10SM ,-,80.1,64,35.7, 58 , 7 ,270,-,0 ,26.30,-,150,AA,-

03013,19960701,1053,AO21,-,CLR ,10SM ,-,82,64.9,35.9, 56 , 3 ,VRB,-,0 ,26.30,6,150,AA,-

03013,19960701,1153,AO21,-,CLR ,10SM ,-,84.9,63,35.8, 48 , 5 ,300,-,0 ,26.29,-,140,AA,-

03013,19960701,1253,AO21,-,CLR ,10SM ,-,88,62.1,35.9, 42 , 3 ,310,-,0 ,26.26,-,132,AA,-

03013,19960701,1353,AO21,-,CLR ,10SM ,-,90,55,35.5, 31 , 3 ,200,-,0 ,26.24,8,123,AA,-

03013,19960701,1453,AO21,-,CLR ,10SM ,-,91,53.1,35.5, 28 , 3 ,200,-,0 ,26.22,-,118,AA,-

03013,19960701,1553,AO21,-,CLR ,10SM ,-,93,55.9,35.7, 29 , 3 ,360,-,0 ,26.21,-,114,AA,-

03013,19960701,1653,AO21,-,CLR ,10SM ,-,91.9,53.1,35.5, 27 , 0 ,000,-,0 ,26.21,6,110,AA,-

03013,19960701,1753,AO21,-,CLR ,10SM ,-,90,60.1,35.8, 37 , 3 ,180,-,0 ,26.20,-,106,AA,-

03013,19960701,1853,AO21,-,CLR ,10SM ,-,82.9,60.1,35.6, 46 , 8 ,170,-,0 ,26.21,-,111,AA,-

03013,19960701,1953,AO21,-,CLR ,10SM ,-,73,63,35.5, 71 , 5 ,180,-,0 ,26.21,3,118,AA,-

03013,19960701,2053,AO22,-,CLR ,10SM ,-,70,64.9,35.5, 84 , 6 ,210,-,0 ,26.22,-,123,AA,-

03013,19960701,2153,AO22,-,CLR ,10SM ,-,68,62.1,35.2, 81 , 6 ,220,-,0 ,26.23,-,123,AA,-

03013,19960701,2253,AO22,-,CLR ,10SM ,-,66.9,62.1,35.2, 84 , 0 ,000,-,0 ,26.24,1,125,AA,-

03013,19960701,2353,AO22,-,CLR ,10SM ,-,66,61,35.1, 84 , 0 ,000,-,0 ,26.24,-,124,AA,-

03013,19960702,0053,AO20,-,CLR ,10SM ,-,64.9,60.1,35, 84 , 0 ,000,-,0 ,26.24,-,123,AA,-

03013,19960702,0153,AO20,-,CLR ,10SM ,-,63,59,34.9, 87 , 4 ,210,-,0 ,26.24,0,123,AA,-

03013,19960702,0253,AO20,-,CLR ,10SM ,-,61,57.9,34.7, 90 , 3 ,270,-,0 ,26.23,-,122,AA,-

03013,19960702,0353,AO20,-,CLR ,10SM ,-,59,55.9,34.5, 90 , 4 ,260,-,0 ,26.24,-,123,AA,-

03013,19960702,0453,AO20,-,CLR ,10SM ,-,59,55.9,34.5, 90 , 0 ,000,-,0 ,26.25,3,128,AA,-

03013,19960702,0553,AO20,-,CLR ,10SM ,-,63,57.9,34.8, 84 , 0 ,000,-,0 ,26.27,-,133,AA,-

03013,19960702,0653,AO20,-,CLR ,10SM ,-,68,61,35.2, 78 , 3 ,280,-,0 ,26.28,-,137,AA,-

03013,19960702,0753,AO20,-,CLR ,10SM ,-,75,59,35.3, 58 , 4 ,270,-,0 ,26.28,1,142,AA,-

03013,19960702,0853,AO20,-,CLR ,10SM ,-,79,59,35.4, 50 , 0 ,000,-,0 ,26.28,-,141,AA,-

03013,19960702,0953,AO20,-,CLR ,10SM ,-,84.9,60.1,35.7, 43 , 0 ,000,-,0 ,26.28,-,137,AA,-

03013,19960702,1053,AO21,-,CLR ,10SM ,-,89.1,57,35.6, 34 , 7 ,120,-,0 ,26.27,8,132,AA,-

03013,19960702,1153,AO21,-,CLR ,10SM ,-,91.9,54,35.5, 28 , 7 ,130,-,0 ,26.25,-,126,AA,-

03013,19960702,1253,AO21,-,CLR ,10SM ,-,93.9,51.1,35.5, 23 , 5 ,110,-,0 ,26.24,-,118,AA,-

03013,19960702,1353,AO21,-,CLR ,10SM ,-,95,55,35.7, 26 , 3 ,VRB,-,0 ,26.22,7,115,AA,-

03013,19960702,1453,AO21,-,CLR ,10SM ,-,96.1,52,35.6, 23 , 4 ,090,-,0 ,26.21,-,110,AA,-

03013,19960702,1553,AO21,-,CLR ,10SM ,-,95,50,35.5, 22 , 6 ,130,-,0 ,26.20,-,108,AA,-

03013,19960702,1653,AO21,-,CLR ,10SM ,-,93.9,52,35.5, 24 , 9 ,120,-,0 ,26.19,7,103,AA,-

03013,19960702,1753,AO21,-,CLR ,10SM ,-,90,55.9,35.6, 32 , 8 ,130,-,0 ,26.20,-,106,AA,-

03013,19960702,1853,AO21,-,CLR ,10SM ,-,84,57,35.5, 40 , 9 ,140,-,0 ,26.21,-,111,AA,-

03013,19960702,1953,AO21,-,CLR ,10SM ,-,81,54,35.2, 39 , 10 ,150,-,0 ,26.23,3,116,AA,-

03013,19960702,2053,AO22,-,CLR ,10SM ,-,78.1,54,35.1, 43 , 11 ,170,-,0 ,26.25,-,123,AA,-

03013,19960702,2153,AO22,-,CLR ,10SM ,-,78.1,53.1,35.1, 42 , 13 ,170,-,0 ,26.25,-,121,AA,-

03013,19960702,2253,AO22,-,CLR ,10SM ,-,75,55.9,35.1, 52 , 10 ,180,-,0 ,26.26,1,125,AA,-

03013,19960702,2353,AO22,-,CLR ,10SM ,-,68,57.9,35, 70 , 5 ,250,-,0 ,26.26,-,126,AA,-

03013,19960703,0053,AO20,-,CLR ,10SM ,-,69.1,57.9,35, 68 , 3 ,260,-,0 ,26.26,-,125,AA,-

03013,19960703,0153,AO20,-,CLR ,10SM ,-,64,59,34.9, 84 , 3 ,340,-,0 ,26.26,0,127,AA,-

03013,19960703,0253,AO20,-,CLR ,10SM ,-,63,57.9,34.8, 84 , 0 ,000,-,0 ,26.26,-,125,AA,-

03013,19960703,0353,AO20,-,CLR ,10SM ,-,61,57.9,34.7, 90 , 0 ,000,-,0 ,26.26,-,127,AA,-

03013,19960703,0453,AO20,-,CLR ,10SM ,-,60.1,57.9,34.7, 93 , 0 ,000,-,0 ,26.26,3,132,AA,-

03013,19960703,0553,AO20,-,CLR ,10SM ,-,68,62.1,35.2, 81 , 3 ,110,-,0 ,26.26,-,129,AA,-

03013,19960703,0653,AO20,-,CLR ,10SM ,-,73.9,60.1,35.3, 62 , 0 ,000,-,0 ,26.27,-,132,AA,-

03013,19960703,0753,AO20,-,CLR ,10SM ,-,79,63,35.6, 58 , 0 ,000,-,0 ,26.27,3,130,AA,-

03013,19960703,0853,AO20,-,CLR ,10SM ,-,84,66,36, 55 , 3 ,VRB,-,0 ,26.26,-,124,AA,-

03013,19960703,0953,AO20,-,CLR ,10SM ,-,89.1,64,36, 43 , 5 ,050,-,0 ,26.25,-,117,AA,-

03013,19960703,1053,AO21,-,CLR ,10SM ,-,91.9,54,35.5, 28 , 6 ,VRB,-,0 ,26.24,6,114,AA,-

03013,19960703,1153,AO21,-,CLR ,10SM ,-,93,57,35.7, 30 , 6 ,200,-,0 ,26.22,-,112,AA,-

03013,19960703,1253,AO21,-,CLR ,10SM ,-,93,57,35.7, 30 , 6 ,230,-,0 ,26.21,-,105,AA,-

03013,19960703,1353,AO21,-,CLR ,10SM ,-,93.9,55,35.7, 27 , 11 ,210,-,0 ,26.18,8,098,AA,-

03013,19960703,1453,AO21,-,CLR ,10SM ,-,95,55.9,35.7, 27 , 12 ,180,-,0 ,26.16,-,089,AA,-

03013,19960703,1553,AO21,-,CLR ,10SM ,-,95,55,35.7, 26 , 11 ,180,-,0 ,26.14,-,083,AA,-

03013,19960703,1653,AO21,-,CLR ,10SM ,-,95,57,35.8, 28 , 12 ,180,-,0 ,26.13,6,079,AA,-

03013,19960703,1753,AO21,-,CLR ,10SM ,-,91.9,57,35.7, 31 , 8 ,160,-,0 ,26.12,-,075,AA,-

03013,19960703,1853,AO21,-,CLR ,10SM ,-,88,57,35.6, 35 , 8 ,150,-,0 ,26.13,-,076,AA,-

03013,19960703,1953,AO21,-,CLR ,10SM ,-,81,57.9,35.4, 46 , 9 ,150,-,0 ,26.14,3,080,AA,-

03013,19960703,2053,AO22,-,CLR ,10SM ,-,80.1,57.9,35.4, 47 , 12 ,150,-,0 ,26.14,-,080,AA,-

03013,19960703,2153,AO22,-,CLR ,10SM ,-,78.1,57,35.3, 48 , 12 ,170,-,0 ,26.14,-,079,AA,-

03013,19960703,2253,AO22,-,CLR ,10SM ,-,75,57,35.2, 54 , 5 ,210,-,0 ,26.15,1,080,AA,-

03013,19960703,2353,AO22,-,CLR ,10SM ,-,70,57.9,35, 66 , 0 ,000,-,0 ,26.14,-,079,AA,-

03013,19960704,0053,AO20,-,CLR ,10SM ,-,68,59,35, 73 , 0 ,000,-,0 ,26.13,-,077,AA,-

03013,19960704,0153,AO20,-,CLR ,10SM ,-,70,57,35, 64 , 3 ,320,-,0 ,26.12,6,073,AA,-

03013,19960704,0253,AO20,-,CLR ,10SM ,-,69.1,55.9,34.9, 63 , 3 ,180,-,0 ,26.12,-,073,AA,-

03013,19960704,0353,AO20,-,CLR ,10SM ,-,62.1,57.9,34.8, 86 , 3 ,260,-,0 ,26.12,-,076,AA,-

03013,19960704,0453,AO20,-,CLR ,10SM ,-,62.1,57,34.7, 84 , 5 ,200,-,0 ,26.12,8,077,AA,-

03013,19960704,0553,AO20,-,CLR ,10SM ,-,66.9,59,35, 76 , 3 ,260,-,0 ,26.13,-,076,AA,-

03013,19960704,0653,AO20,-,CLR ,10SM ,-,73,63,35.5, 71 , 0 ,000,-,0 ,26.13,-,077,AA,-

03013,19960704,0753,AO20,-,CLR ,10SM ,-,81,64,35.8, 57 , 5 ,310,-,0 ,26.13,0,075,AA,-

03013,19960704,0853,AO20,-,CLR ,10SM ,-,86,63,35.8, 46 ,
7 ,300,-,0 ,26.12,-,072,AA,-
03013,19960704,0953,AO20,-,CLR ,10SM ,-,89.1,62.1,35.9,
41 , 7 ,300,-,0 ,26.12,-,069,AA,-
03013,19960704,1053,AO21,-,CLR ,10SM ,-,91.9,60.1,35.9,
34 , 3 ,VRB,-,0 ,26.12,8,067,AA,-
03013,19960704,1153,AO21,-,CLR ,10SM ,-,95,62.1,36, 34 ,
4 ,070,-,0 ,26.11,-,065,AA,-
03013,19960704,1253,AO21,-,CLR ,10SM ,-,99,54,35.7, 22 ,
6 ,120,-,0 ,26.10,-,060,AA,-
03013,19960704,1353,AO21,-,CLR ,10SM ,-,99,53.1,35.7, 21
, 6 ,090,-,0 ,26.08,8,053,AA,-
03013,19960704,1453,AO21,-,CLR ,10SM ,-,100.9,51.1,35.7,
19 , 0 ,000,-,0 ,26.06,-,046,AA,-
03013,19960704,1553,AO21,-,CLR ,10SM ,-,99,51.1,35.6, 20
, 10 ,140,-,0 ,26.05,-,045,AA,-
03013,19960704,1653,AO21,-,CLR ,10SM ,-,98.1,51.1,35.6,
20 , 8 ,130,-,0 ,26.04,6,044,AA,-
03013,19960704,1753,AO21,-,CLR ,10SM ,-,93.9,57.9,35.8,
30 , 6 ,130,-,0 ,26.05,-,045,AA,-
03013,19960704,1853,AO21,-,CLR ,10SM ,-,87.1,54,35.4, 32
, 9 ,140,-,0 ,26.05,-,047,AA,-
03013,19960704,1953,AO21,-,CLR ,10SM ,-,86,51.1,35.2, 30
, 13 ,170,-,0 ,26.05,1,044,AA,-
03013,19960704,2053,AO22,-,CLR ,10SM ,-,84,53.1,35.3, 35
, 12 ,180,-,0 ,26.07,-,047,AA,-
03013,19960704,2153,AO22,-,CLR ,10SM ,-,82.9,52,35.2, 34
, 14 ,180,-,0 ,26.08,-,050,AA,-
03013,19960704,2253,AO22,-,CLR ,10SM ,-,81,52,35.1, 37 ,
9 ,200,-,0 ,26.09,1,054,AA,-
03013,19960704,2353,AO22,-,CLR ,10SM ,-,78.1,55,35.2, 45
, 6 ,070,-,0 ,26.12,-,065,AA,-
03013,19960705,0053,AO20,-,CLR ,10SM ,-,79,61,35.5, 54 ,
13 ,050,-,0 ,26.12,-,065,AA,-
03013,19960705,0153,AO20,-,CLR ,10SM ,-,78.1,62.1,35.5,
58 , 10 ,090,-,0 ,26.11,0,060,AA,-
03013,19960705,0253,AO20,-,CLR ,10SM ,-,75,63,35.5, 66 ,
9 ,090,-,0 ,26.10,-,055,AA,-
03013,19960705,0353,AO20,-,CLR ,10SM ,-,75,64.9,35.6, 71
, 10 ,130,-,0 ,26.11,-,060,AA,-
03013,19960705,0453,AO20,-,CLR ,10SM ,-,73.9,64.9,35.6,
74 , 8 ,120,-,0 ,26.12,5,064,AA,-

03013,19960705,0553,AO20,-,CLR ,10SM ,-,73.9,64.9,35.6, 74 , 7 ,170,-,0 ,26.13,-,074,AA,-

03013,19960705,0653,AO20,-,CLR ,10SM ,-,75.9,68,35.9, 77 , 9 ,050,-,0 ,26.16,-,085,AA,-

03013,19960705,0753,AO20,-,CLR ,10SM ,-,78.1,68,35.9, 71 , 11 ,070,-,0 ,26.15,0,083,AA,-

03013,19960705,0853,AO20,-,CLR ,10SM ,-,81,68,36, 65 , 9 ,100,-,0 ,26.15,-,080,AA,-

03013,19960705,0953,AO20,-,CLR ,10SM ,-,84,66.9,36, 57 , 5 ,130,-,0 ,26.15,-,080,AA,-

03013,19960705,1053,AO21,-,CLR ,10SM ,-,87.1,66.9,36.1, 51 , 8 ,080,-,0 ,26.14,6,077,AA,-

03013,19960705,1153,AO21,-,CLR ,10SM ,-,90,,34.6, 13 , 6 ,060,-,0 ,26.12,-,071,AA,-

03013,19960705,1253,AO21,-,CLR ,10SM ,-,93,66.9,36.2, 42 , 10 ,060,-,0 ,26.11,-,061,AA,-

03013,19960705,1353,AO21,-,CLR ,10SM ,-,96.1,64.9,36.2, 36 , 3 ,VRB,-,0 ,26.09,8,053,AA,-

03013,19960705,1453,AO21,-,CLR ,10SM ,-,99,60.1,36, 28 , 0 ,000,-,0 ,26.07,-,046,AA,-

03013,19960705,1553,AO21,-,CLR ,10SM ,-,93.9,71.1,36.5, 48 , 7 ,360,-,0 ,26.06,-,045,AA,-

03013,19960705,1653,AO21,-,CLR ,10SM ,-,84.9,59,35.6, 42 , 20 ,360,G,25 ,26.10,5,062,AA,-

03013,19960705,1753,AO21,-,CLR ,10SM ,-,84,57,35.5, 40 , 13 ,010,-,0 ,26.12,-,074,AA,-

03013,19960705,1853,AO21,-,FEW065 SCT110 ,10SM ,-RA ,81,61,35.6, 51 , 12 ,020,-,0 ,26.13,-,079,AA, T

03013,19960705,1953,AO21,-,FEW110 ,10SM ,-,78.1,61,35.5, 56 , 14 ,200,-,0 ,26.19,3,102,AA, T

03013,19960705,2053,AO22,-,CLR ,10SM ,-,73.9,62.1,35.4, 67 , 14 ,130,-,0 ,26.16,-,090,AA,-

03013,19960705,2153,AO22,-,CLR ,10SM ,-,73.9,64,35.5, 71 , 4 ,070,-,0 ,26.14,-,081,AA,-

03013,19960705,2253,AO22,-,CLR ,10SM ,-,71.1,64,35.5, 79 , 7 ,270,-,0 ,26.16,5,090,AA,-

03013,19960705,2353,AO22,-,CLR ,10SM ,-,72,64,35.5, 76 , 11 ,240,-,0 ,26.21,-,105,AA,-

03013,19960706,0053,AO20,-,CLR ,10SM ,-,72,61,35.3, 69 , 7 ,320,-,0 ,26.19,-,097,AA,-

03013,19960706,0153,AO20,-,CLR ,10SM ,-,70,57.9,35, 66 , 6 ,350,-,0 ,26.18,0,092,AA,-

03013,19960706,0253,AO20,-,CLR ,10SM ,-,68,57.9,35, 70 ,
0 ,000,-,0 ,26.17,-,087,AA,-
03013,19960706,0353,AO20,-,CLR ,10SM ,-,63,59,34.9, 87 ,
7 ,200,-,0 ,26.17,-,092,AA,-
03013,19960706,0453,AO20,-,CLR ,10SM ,-,63,59,34.9, 87 ,
6 ,220,-,0 ,26.19,3,105,AA,-
03013,19960706,0553,AO20,-,CLR ,10SM ,-,71.1,59,35.2, 66
, 8 ,260,-,0 ,26.21,-,109,AA,-
03013,19960706,0653,AO20,-,CLR ,10SM ,-,75,60.1,35.3, 60
, 7 ,290,-,0 ,26.22,-,116,AA,-
03013,19960706,0753,AO20,-,CLR ,10SM ,-,77,60.1,35.4, 56
, 7 ,290,-,0 ,26.23,1,117,AA,-
03013,19960706,0853,AO20,-,CLR ,10SM ,-,81,60.1,35.5, 49
, 7 ,290,-,0 ,26.22,-,113,AA,-
03013,19960706,0953,AO20,-,CLR ,10SM ,-,86,57.9,35.6, 39
, 6 ,270,-,0 ,26.21,-,109,AA,-
03013,19960706,1053,AO21,-,CLR ,10SM ,-,88,59,35.7, 38 ,
3 ,210,-,0 ,26.21,6,106,AA,-
03013,19960706,1153,AO21,-,CLR ,10SM ,-,91,61,35.9, 37 ,
4 ,VRB,-,0 ,26.21,-,103,AA,-
03013,19960706,1253,AO21,-,CLR ,10SM ,-,93.9,60.1,35.9,
32 , 3 ,VRB,-,0 ,26.20,-,099,AA,-
03013,19960706,1353,AO21,-,CLR ,10SM ,-,96.1,60.1,36, 30
, 5 ,VRB,-,0 ,26.19,8,095,AA,-
03013,19960706,1453,AO21,-,CLR ,10SM ,-,97,55,35.7, 25 ,
9 ,040,-,0 ,26.17,-,088,AA,-
03013,19960706,1553,AO21,-,CLR ,10SM ,-,97,53.1,35.7, 23
, 11 ,050,-,0 ,26.16,-,086,AA,-
03013,19960706,1653,AO21,-,CLR ,10SM ,-,96.1,57,35.8, 27
, 9 ,100,-,0 ,26.16,5,086,AA,-
03013,19960706,1753,AO21,-,CLR ,10SM ,-,93,57.9,35.8, 31
, 8 ,120,-,0 ,26.17,-,089,AA,-
03013,19960706,1853,AO21,-,CLR ,10SM ,-,86,60.1,35.7, 42
, 7 ,140,-,0 ,26.18,-,094,AA,-
03013,19960706,1953,AO21,-,CLR ,10SM ,-,81,59,35.5, 47 ,
7 ,150,-,0 ,26.20,3,103,AA,-
03013,19960706,2053,AO22,-,CLR ,10SM ,-,75.9,60.1,35.4,
58 , 0 ,000,-,0 ,26.21,-,106,AA,-
03013,19960706,2153,AO22,-,CLR ,10SM ,-,73,64,35.5, 74 ,
5 ,010,-,0 ,26.21,-,108,AA,-
03013,19960706,2253,AO22,-,CLR ,10SM ,-,78.1,64,35.7, 62
, 11 ,020,-,0 ,26.22,3,111,AA,-

03013,19960706,2353,AO22,-,CLR ,10SM ,-,73.9,57.9,35.2, 57 , 10 ,030,-,0 ,26.25,-,119,AA,-

03013,19960707,0053,AO20,-,CLR ,10SM ,-,73,57,35.1, 57 , 9 ,060,-,0 ,26.25,-,119,AA,-

03013,19960707,0153,AO20,-,CLR ,10SM ,-,73,59,35.2, 62 , 14 ,100,G,19 ,26.24,0,114,AA,-

03013,19960707,0253,AO20,-,CLR ,10SM ,-,69.1,57.9,35, 68 , 8 ,120,-,0 ,26.22,-,110,AA,-

03013,19960707,0353,AO20,-,CLR ,10SM ,-,69.1,61,35.2, 76 , 5 ,130,-,0 ,26.22,-,110,AA,-

03013,19960707,0453,AO20,-,CLR ,10SM ,-,66,61,35.1, 84 , 0 ,000,-,0 ,26.23,5,114,AA,-

03013,19960707,0553,AO20,-,BKN021 ,10SM ,-,66.9,61,35.1, 81 , 4 ,310,-,0 ,26.26,-,126,AA,-

03013,19960707,0615,AO20,-,SCT021 ,10SM ,-,-,-,-,-, 4 ,290,- ,0 ,-,-,-,SP,-

03013,19960707,0653,AO20,-,SCT024 ,10SM ,-,72,63,35.4, 73 , 3 ,350,-,0 ,26.28,-,135,AA,-

03013,19960707,0735,AO20,-,BKN026 ,10SM ,-,-,-,-,-, 5 ,030,-,0 ,-,-,-,SP,-

03013,19960707,0753,AO20,-,OVC029 ,10SM ,-,73.9,63,35.5, 69 , 5 ,050,-,0 ,26.29,1,140,AA,-

03013,19960707,0806,AO20,-,BKN031 ,10SM ,-,-,-,-,-, 5 ,050,-,0 ,-,-,-,SP,-

03013,19960707,0853,AO20,-,SCT033 ,10SM ,-,78.1,62.1,35.6, 58 , 8 ,120,-,0 ,26.27,-,135,AA,-

03013,19960707,0953,AO20,-,CLR ,10SM ,-,82,64,35.8, 55 , 11 ,150,-,0 ,26.25,-,124,AA,-

03013,19960707,1053,AO21,-,CLR ,10SM ,-,86,63,35.8, 46 , 11 ,170,-,0 ,26.22,8,110,AA,-

03013,19960707,1129,AO21,-,CLR ,10SM ,-,-,-,-,-, 13 ,180,G,18 ,-,-,-,SP,-

03013,19960707,1153,AO21,-,CLR ,10SM ,-,88,64,36, 45 , 12 ,150,-,0 ,26.20,-,100,AA,-

03013,19960707,1253,AO21,-,CLR ,10SM ,-,91,66,36.2, 44 , 9 ,170,G,18 ,26.16,-,087,AA,-

03013,19960707,1353,AO21,-,CLR ,10SM ,-,93.9,64.9,36.2, 38 , 11 ,170,G,15 ,26.13,8,073,AA,-

03013,19960707,1415,AO21,-,CLR ,10SM ,-,-,-,-,-, 12 ,150,- ,0 ,-,-,-,SP,-

03013,19960707,1427,AO21,-,CLR ,10SM ,-,-,-,-,-, 13 ,170,G,20 ,-,-,-,SP,-

03013,19960707,1441,AO21,-,CLR ,10SM ,-,-,-,-,-, 12 ,160,-
,0 ,-,-,-,SP,-

03013,19960707,1453,AO21,-,CLR ,10SM ,-,95,63,36.1, 35 ,
12 ,190,-,0 ,26.10,-,062,AA,-

03013,19960707,1553,AO21,-,CLR ,10SM ,-,93.9,60.1,35.9,
32 , 11 ,190,-,0 ,26.08,-,055,AA,-

03013,19960707,1653,AO21,-,CLR ,10SM ,-,93,61,35.9, 34 ,
9 ,160,-,0 ,26.05,6,047,AA,-

03013,19960707,1716,AO21,-,SCT090 ,10SM ,-,-,-,-,-, 11
,230,-,0 ,-,-,-,SP,-

03013,19960707,1753,AO21,-,-,10SM ,-RA ,77,62.1,35.3, 60 , 12 ,140,-,0 ,
.00,-,-,AA,.07

03013,19960707,1814,AO21,-,SCT065 BKN110 ,10SM ,-RA SQ
,-,-,-,-, 21 ,310,G,40 ,-,-,-,SP,-

03013,19960707,1828,AO21,-,SCT075 BKN110 ,10SM ,-RA
,-,-,-,-, 12 ,350,-,0 ,-,-,-,SP,-

03013,19960707,1853,AO21,-,BKN060 BKN100 ,10SM ,-RA
,75,61,35.2, 62 , 6 ,240,-,0 , .00,-,-,AA,.09

03013,19960707,1943,AO21,-,BKN050 BKN065 OVC110 ,10SM ,-RA
,-,-,-,-, 21 ,200,-,0 ,-,-,-,SP,-

03013,19960707,1953,AO21,-,BKN050 OVC110 ,10SM ,-RA
,73,60.1,35, 64 , 31 ,180,-,0 , .00,-,-,AA,.01

03013,19960707,2053,AO22,-,CLR ,10SM ,-,75.9,55.9,35.2,
50 , 14 ,160,-,0 ,26.07,-,057,AA,-

03013,19960708,0053,AO20,-,CLR ,10SM ,-,62.1,62.1,35,100
, 3 ,040,-,0 ,26.12,-,077,AA,-

03013,19960708,0153,AO20,-,CLR ,10SM ,-,63,62.1,35, 97 ,
5 ,080,-,0 ,26.12,5,074,AA,-

03013,19960708,0253,AO20,-,CLR ,8SM ,-,61,61,34.9,100 ,
8 ,040,-,0 ,26.14,-,083,AA,-

03013,19960708,0353,AO20,-,CLR ,10SM ,-,62.1,61,34.9, 96
, 7 ,070,-,0 ,26.14,-,081,AA,-

03013,19960708,0453,AO20,-,CLR ,10SM ,-,62.1,61,34.9, 96
, 7 ,090,-,0 ,26.16,2,089,AA,-

03013,19960708,0553,AO20,-,CLR ,9SM ,-,64,63,35.1, 96 , 6
,060,-,0 ,26.21,-,118,AA,-

03013,19960708,0635,AO20,-,BKN010 ,10SM ,-,-,-,-,-, 14
,070,-,0 ,-,-,-,SP,-

03013,19960708,0645,AO20,-,OVC008 ,10SM ,-,-,-,-,-, 13
,080,G,23 ,-,-,-,SP,-

03013,19960708,0653,AO20,-,OVC008 ,10SM ,-,64.9,63,35.2,
93 , 13 ,080,-,0 ,26.24,-,131,AA,-

03013,19960708,0722,AO20,-,OVC006 ,7SM ,-,-,-,-,-, 11
,060,-,0 ,-,-,-,SP,-
03013,19960708,0736,AO20,-,OVC008 ,8SM ,-,-,-,-,-, 9
,060,-,0 ,-,-,-,SP,-
03013,19960708,0753,AO20,-,OVC008 ,10SM ,-,68,64,35.3, 87
, 10 ,070,-,0 ,26.25,1,136,AA,-
03013,19960708,0816,AO20,-,OVC010 ,10SM ,-,-,-,-,-, 10
,060,-,0 ,-,-,-,SP,-
03013,19960708,0853,AO20,-,OVC012 ,10SM ,-,69.1,64,35.4,
84 , 5 ,060,-,0 ,26.27,-,138,AA,-
03013,19960708,0930,AO20,-,SCT016 OVC023 ,10SM ,-,-,-,-,-, 5
,080,-,0 ,-,-,-,SP,-
03013,19960708,0953,AO20,-,BKN023 BKN031 OVC038 ,10SM ,-
,72,64.9,35.5, 79 , 8 ,070,-,0 ,26.28,-,143,AA,-
03013,19960708,1001,AO21,-,SCT023 BKN031 OVC040 ,10SM ,-,-,-,-
,-, 4 ,VRB,-,0 ,-,-,-,SP,-
03013,19960708,1018,AO21,-,BKN021 OVC040 ,10SM ,-,-,-,-,-, 10
,070,-,0 ,-,-,-,SP,-
03013,19960708,1029,AO21,-,SCT021 OVC040 ,10SM ,-,-,-,-,-, 10
,090,-,0 ,-,-,-,SP,-
03013,19960708,1053,AO21,-,SCT027 OVC040 ,10SM ,-
,75,64,35.6, 69 , 13 ,060,G,16 ,26.27,0,143,AA,-
03013,19960708,1104,AO21,-,BKN027 OVC038 ,10SM ,-,-,-,-,-, 11
,070,-,0 ,-,-,-,SP,-
03013,19960708,1111,AO21,-,SCT027 OVC038 ,10SM ,-,-,-,-,-, 10
,060,-,0 ,-,-,-,SP,-
03013,19960708,1153,AO21,-,BKN030 OVC038 ,10SM ,-
,73,64,35.5, 74 , 7 ,040,-,0 ,26.28,-,149,AA,-
03013,19960708,1208,AO21,-,OVC027 ,10SM ,-,-,-,-,-, 8
,040,-,0 ,-,-,-,SP,-
03013,19960708,1253,AO21,-,OVC027 ,10SM ,-,72,63,35.4, 73
, 11 ,080,-,0 ,26.28,-,150,AA,-
03013,19960708,1319,AO21,-,SCT027 OVC039 ,10SM ,-,-,-,-,-, 9
,100,-,0 ,-,-,-,SP,-
03013,19960708,1353,AO21,-,SCT039 OVC060 ,10SM ,-
,73.9,64,35.5, 71 , 10 ,070,-,0 ,26.25,8,139,AA,-
03013,19960708,1453,AO21,-,OVC070 ,10SM ,-,75,64,35.6, 69
, 6 ,110,-,0 ,26.24,-,135,AA,-
03013,19960708,1553,AO21,-,BKN060 ,10SM ,-,75,64.9,35.6,
71 , 5 ,080,-,0 ,26.24,-,134,AA,-
03013,19960708,1653,AO21,-,SCT055 ,10SM ,-,75,64.9,35.6,
71 , 10 ,090,-,0 ,26.23,6,130,AA,-

03013,19960708,1753,AO21,-,FEW055 OVC075 ,10SM ,-
,73,66,35.6, 79 , 7 ,100,-,0 ,26.23,-,131,AA,-

03013,19960708,1853,AO21,-,BKN049 BKN070 ,10SM ,-
,72,64.9,35.5, 79 , 11 ,100,-,0 ,26.24,-,134,AA,-

03013,19960708,1953,AO21,-,BKN037 OVC046 ,10SM ,-
,71.1,64.9,35.5, 81 , 8 ,090,-,0 ,26.25,3,134,AA,-

03013,19960708,2053,AO22,-,SCT060 BKN095 ,10SM ,-
,69.1,64,35.4, 84 , 14 ,120,-,0 ,26.25,-,139,AA,-

03013,19960708,2153,AO22,-,OVC100 ,10SM ,-,68,64,35.3, 87
, 9 ,090,-,0 ,26.27,-,144,AA,-

03013,19960708,2210,AO22,-,BKN022 OVC095 ,10SM ,-,-,-,-,-,-, 8
,090,-,0 ,-,-,-,SP,-

03013,19960708,2253,AO22,-,OVC018 ,10SM ,-,68,64,35.3, 87
, 6 ,110,-,0 ,26.28,1,143,AA,-

03013,19960708,2353,AO22,-,OVC018 ,10SM ,-,66.9,64,35.3,
91 , 9 ,140,-,0 ,26.27,-,144,AA,-

03013,19960709,0039,AO20,-,SCT015 SCT024 BKN100 ,10SM ,-,-,-,-,-
, 10 ,120,-,0 ,-,-,-,SP,-

03013,19960709,0053,AO20,-,FEW015 BKN100 ,10SM ,-
,66,64,35.3, 93 , 8 ,120,-,0 ,26.26,-,142,AA,-

03013,19960709,0137,AO20,-,BKN026 OVC100 ,10SM ,-,-,-,-,-,-, 7
,120,-,0 ,-,-,-,SP,-

03013,19960709,0153,AO20,-,OVC023 ,10SM ,-,66,64,35.3, 93
, 9 ,130,-,0 ,26.26,8,139,AA,-

03013,19960709,0221,AO20,-,FEW019 BKN050 OVC100 ,10SM ,-,-,-,-
,-, 7 ,130,-,0 ,-,-,-,SP,-

03013,19960709,0253,AO20,-,SCT023 OVC029 ,10SM ,-
,66,64,35.3, 93 , 6 ,090,-,0 ,26.27,-,140,AA,-

03013,19960709,0333,AO20,-,BKN014 OVC024 ,8SM ,-,-,-,-,-,-, 9
,110,-,0 ,-,-,-,SP,-

03013,19960709,0353,AO20,-,OVC012 ,8SM ,-,66,64,35.3, 93
, 7 ,100,-,0 ,26.28,-,144,AA,-

03013,19960709,0400,AO20,-,BKN008 OVC014 ,8SM ,-,-,-,-,-,-, 7
,110,-,0 ,-,-,-,SP,-

03013,19960709,0410,AO20,-,BKN006 OVC012 ,8SM ,-,-,-,-,-,-, 9
,100,-,0 ,-,-,-,SP,-

03013,19960709,0453,AO20,-,SCT006 OVC012 ,10SM ,-
,64.9,62.1,35.1, 90 , 10 ,130,-,0 ,26.29,3,148,AA,-

03013,19960709,0536,AO20,-,SCT010 BKN021 OVC031 ,10SM ,-,-,-,-
,-, 11 ,130,-,0 ,-,-,-,SP,-

03013,19960709,0545,AO20,-,FEW010 SCT022 OVC031 ,10SM ,-,-,-,-
,-, 9 ,140,-,0 ,-,-,-,SP,-

03013,19960709,0553,AO20,-,FEW022 OVC031 ,10SM ,-
,66,63,35.2, 90 , 9 ,150,-,0 ,26.29,-,151,AA,-

03013,19960709,0611,AO20,-,BKN018 BKN029 OVC034 ,8SM ,-RA
,-,-,-,-, 7 ,140,-,0 ,-,-,-,SP,-

03013,19960709,0634,AO20,-,OVC013 ,10SM ,-,-,-,-,-, 7
,130,-,0 ,-,-,-,SP,-

03013,19960709,0653,AO20,-,BKN013 OVC018 ,10SM ,-
,66,63,35.2, 90 , 7 ,130,-,0 ,26.30,-,159,AA, T

03013,19960709,0707,AO20,-,OVC015 ,10SM ,-,-,-,-,-, 9
,140,-,0 ,-,-,-,SP,-

03013,19960709,0753,AO20,-,OVC017 ,10SM ,-,66,62.1,35.2,
87 , 8 ,090,-,0 ,26.31,3,161,AA,-

03013,19960709,0853,AO20,-,BKN023 OVC036 ,10SM ,-
,70,62.1,35.3, 76 , 11 ,160,-,0 ,26.31,-,158,AA,-

03013,19960709,0906,AO20,-,SCT023 BKN036 BKN065 ,10SM ,-,-,-,-
,-, 9 ,150,-,0 ,-,-,-,SP,-

03013,19960709,0924,AO20,-,SCT020 BKN026 OVC065 ,10SM ,-,-,-,-
,-, 10 ,120,-,0 ,-,-,-,SP,-

03013,19960709,0953,AO20,-,FEW009 BKN020 OVC026 ,10SM ,-
,66.9,62.1,35.2, 84 , 13 ,090,-,0 ,26.31,-,162,AA,-

03013,19960709,1019,AO21,-,BKN009 BKN019 OVC027 ,10SM ,-,-,-,-
,-, 13 ,110,-,0 ,-,-,-,SP,-

03013,19960709,1053,AO21,-,OVC005 ,6SM ,BR
,64,62.1,35.1, 93 , 13 ,100,-,0 ,26.31,3,166,AA,-

03013,19960709,1109,AO21,-,OVC003 ,7SM ,-,-,-,-,-, 14
,110,-,0 ,-,-,-,SP,-

03013,19960709,1136,AO21,-,OVC005 ,10SM ,-,-,-,-,-, 13
,110,-,0 ,-,-,-,SP,-

03013,19960709,1153,AO21,-,OVC005 ,10SM ,-,64,62.1,35.1,
93 , 12 ,120,-,0 ,26.32,-,168,AA,-

03013,19960709,1215,AO21,-,OVC007 ,10SM ,-,-,-,-,-, 13
,120,-,0 ,-,-,-,SP,-

03013,19960709,1253,AO21,-,OVC007 ,10SM ,-
,64.9,62.1,35.1, 90 , 14 ,110,-,0 ,26.31,-,164,AA, T

03013,19960709,1353,AO21,-,OVC007 ,10SM ,-,64,62.1,35.1,
93 , 14 ,120,-,0 ,26.31,4,167,AA,-

03013,19960709,1442,AO21,-,OVC005 ,7SM ,-,-,-,-,-, 16
,120,-,0 ,-,-,-,SP,-

03013,19960709,1453,AO21,-,OVC005 ,6SM ,BR
,63,61,35, 93 , 16 ,110,-,0 ,26.30,-,160,AA,-

03013,19960709,1522,AO21,-,OVC003 ,5SM ,BR ,-
,-,-,-, 14 ,120,-,0 ,-,-,-,SP,-

03013,19960709,1553,AO21,-,OVC003 ,8SM ,-,63,61,35, 93 , 14 ,110,-,0 ,26.30,-,162,AA,-

03013,19960709,1653,AO21,-,OVC003 ,6SM ,BR ,61,60.1,34.8, 97 , 15 ,120,-,0 ,26.30,5,167,AA,-

03013,19960709,1715,AO21,-,OVC003 ,2 1/2SM,BR ,-,-,-,-, 11 ,110,-,0 ,-,-,-,SP,-

03013,19960709,1726,AO21,-,OVC003 ,3SM ,BR ,- ,-,-,-, 12 ,100,-,0 ,-,-,-,SP,-

03013,19960709,1753,AO21,-,OVC003 ,6SM ,BR ,61,60.1,34.8, 97 , 10 ,100,-,0 ,26.32,-,173,AA,-

03013,19960709,1823,AO21,-,OVC005 ,10SM ,-,-,-,-,-, 12 ,110,-,0 ,-,-,-,SP,-

03013,19960709,1853,AO21,-,OVC005 ,10SM ,-,61,59,34.8, 93 , 9 ,130,-,0 ,26.35,-,182,AA,-

03013,19960709,1938,AO21,-,OVC007 ,10SM ,-RA ,-,-,-,-, 11 ,110,-,0 ,-,-,-,SP,-

03013,19960709,1953,AO21,-,-,5SM ,-RA BR ,61,60.1,34.8, 97 , 10 ,110,-,0 ,26.36,-,184,AA,-

03013,19960709,2016,AO22,-,BKN008 OVC012 ,10SM ,-RA ,-,-,-,-, 9 ,120,-,0 ,-,-,-,SP,-

03013,19960709,2043,AO22,-,OVC006 ,9SM ,-RA ,- ,-,-,-, 9 ,110,-,0 ,-,-,-,SP,-

03013,19960709,2053,AO22,-,OVC006 ,9SM ,-RA ,60.1,60.1,34.8,100 , 10 ,110,-,0 ,26.38,-,191,AA,.04

03013,19960709,2153,AO22,-,BKN004 OVC008 ,7SM ,-RA ,60.1,60.1,34.8,100 , 9 ,100,-,0 ,26.38,-,194,AA,.05

03013,19960709,2225,AO22,-,OVC006 ,9SM ,-RA ,- ,-,-,-, 10 ,110,-,0 ,-,-,-,SP,-

03013,19960709,2253,AO22,-,BKN006 OVC010 ,8SM ,- ,60.1,60.1,34.8,100 , 7 ,110,-,0 ,26.37,0,191,AA, T

03013,19960725,0753,AO20,-,CLR ,10SM ,-,73.9,59,35.2, 60 , 19 ,200,G,25 ,26.40,8,189,AA,-

03013,19960725,0853,AO20,-,FEW075 ,10SM ,-RA ,71.1,62.1,35.3, 73 , 14 ,210,-,0 ,26.39,-,190,AA, T

03013,19960725,0953,AO20,-,CLR ,10SM ,-,79,59,35.4, 50 , 22 ,210,G,28 ,26.37,-,176,AA, T

03013,19960725,1053,AO21,-,CLR ,10SM ,-,80.1,60.1,35.5, 51 , 22 ,210,G,27 ,26.36,8,170,AA,-

03013,19960710,0153,AO20,-,BKN006 OVC011 ,8SM ,- ,60.1,60.1,34.8,100 , 8 ,110,-,0 ,26.37,0,190,AA,-

03013,19960710,0218,AO20,-,BKN004 OVC009 ,8SM ,-,-,-,-,-, 8 ,120,-,0 ,-,-,-,SP,-

03013,19960710,0246,AO20,-,OVC006 ,10SM ,-RA
,-,-,-,-, 9 ,140,-,0 ,-,-,-,SP,-
03013,19960710,0253,AO20,-,BKN006 OVC012 ,10SM ,-RA
,60.1,59,34.7, 96 , 10 ,140,-,0 ,26.36,-,187,AA, T
03013,19960710,0305,AO20,-,BKN008 OVC014 ,9SM ,-RA
,-,-,-,-, 6 ,110,-,0 ,-,-,-,SP,-
03013,19960710,0325,AO20,-,OVC005 ,6SM ,-RA BR
,-,-,-,-, 8 ,120,-,0 ,-,-,-,SP,-
03013,19960710,0353,AO20,-,BKN005 BKN009 OVC013 ,4SM ,-RA
BR ,59,57.9,34.6, 96 , 7 ,120,-,0 ,26.36,-,189,AA,.03
03013,19960710,0431,AO20,-,BKN007 BKN017 OVC034 ,9SM ,-RA
,-,-,-,-, 7 ,120,-,0 ,-,-,-,SP,-
03013,19960710,0453,AO20,-,BKN007 OVC010 ,10SM ,-
,59,57.9,34.6, 96 , 8 ,120,-,0 ,26.36,6,190,AA,.01
03013,19960710,0553,AO20,-,BKN007 BKN010 OVC025 ,5SM ,-RA
BR ,59,57.9,34.6, 96 , 8 ,130,-,0 ,26.37,-,193,AA,.01
03013,19960710,0603,AO20,-,BKN004 OVC009 ,5SM ,-RA BR
,-,-,-,-, 9 ,110,-,0 ,-,-,-,SP,-
03013,19960710,0631,AO20,-,BKN006 BKN009 OVC015 ,10SM ,-,-,-,-
,-, 12 ,120,-,0 ,-,-,-,SP,-
03013,19960710,0645,AO20,-,SCT006 BKN012 OVC021 ,10SM ,-,-,-,-
,-, 9 ,120,-,0 ,-,-,-,SP,-
03013,19960710,0653,AO20,-,BKN006 BKN012 OVC021 ,10SM ,-
,57.9,57,34.5, 97 , 8 ,120,-,0 ,26.38,-,197,AA, T
03013,19960710,0753,AO20,-,OVC006 ,10SM ,-,57.9,57,34.5,
97 , 11 ,120,-,0 ,26.38,1,201,AA,-
03013,19960710,0820,AO20,-,OVC008 ,10SM ,-,-,-,-,-,-, 14
,130,-,0 ,-,-,-,SP,-
03013,19960710,0853,AO20,-,OVC008 ,10SM ,-,59,57,34.6, 93
, 10 ,120,-,0 ,26.38,-,200,AA, T
03013,19960710,0917,AO20,-,OVC006 ,8SM ,-,-,-,-,-,-, 10
,120,-,0 ,-,-,-,SP,-

APPENDIX D

The Analysis Dataset

Year	Humidity	Visibility	Temp
20000101	128	29	26
20000102	155	30	27
20000103	164	30	27
20000104	207	33	29
20000105	181	37	32
20000106	146	37	34
20000107	159	36	32
20000108	164	38	34
20000109	159	43	38
20000110	177	43	39
20000111	183	41	37
20000112	157	40	35
20000113	171	39	34
20000114	170	38	34
20000115	152	37	32
20000116	165	36	31
20000117	156	35	31
20000118	162	33	30
20000119	170	34	30
20000120	183	35	31
20000121	184	39	34
20000122	154	41	36
20000123	149	43	39
20000124	165	41	38
20000125	179	41	35
20000126	194	42	35
20000127	156	42	37
20000128	143	43	39
20000129	133	42	39
20000130	152	41	38
20000131	179	39	55
20000301	171	37	34
20000302	173	34	30
20000303	174	35	31
20000304	174	33	29
20000305	164	32	27
20000306	149	35	31
20000307	148	38	34
20000308	157	40	35
20000309	169	40	35
20000310	155	40	35
20000311	164	38	32
20000312	182	38	32
20000313	152	37	31

20000314	144	39	33
20000315	147	42	36
20000316	153	42	37
20000317	171	41	35
20000318	140	42	36
20000319	152	44	39
20000320	144	47	42
20000321	122	45	40
20000322	118	43	37
20000323	129	42	36
20000324	149	42	36
20000325	166	40	35
20000326	185	36	31
20000327	194	32	27
20000328	188	33	37
20000329	173	35	30
20000330	166	36	32
20000331	148	34	30
20000401	151	32	26
20000402	169	36	30
20000403	189	42	36
20000404	216	44	38
20000405	202	46	41
20000406	186	44	39
20000407	164	39	32
20000408	208	42	35
20000409	196	46	40
20000410	155	47	41
20000411	139	47	41
20000412	158	46	41
20000413	142	43	39
20000414	144	43	38
20000415	137	44	40
20000416	145	45	41
20000417	146	45	40
20000418	154	40	35
20000419	141	36	29
20000420	155	42	35
20000421	179	44	38
20000422	178	42	37
20000423	171	43	39
20000424	159	46	40
20000425	145	49	42
20000426	138	52	46
20000427	157	52	46
20000428	164	50	56
20000429	143	50	43

20000430	165	49	42
20000501	167	46	40
20000502	156	45	37
20000503	137	46	39
20000504	153	47	40
20000505	162	49	42
20000506	158	53	47
20000507	164	55	49
20000508	185	55	49
20000509	188	57	50
20000510	177	57	52
20000511	185	59	54
20000512	189	61	54
20000513	185	62	55
20000514	172	63	56
20000515	157	63	55
20000516	151	62	55
20000517	149	59	52
20000518	163	55	49
20000519	148	53	46
20000520	144	53	46
20000521	139	54	46
20000522	160	55	47
20000523	180	52	45
20000524	188	51	45
20000525	184	53	47
20000526	167	56	50
20000527	168	56	49
20000528	161	57	51
20000529	142	57	50
20000530	139	56	50
20000531	142	55	48
20000601	161	56	49
20000602	170	60	52
20000603	149	61	54
20000604	128	61	54
20000605	139	60	53
20000606	157	58	51
20000607	160	59	51
20000608	163	59	52
20000609	168	60	54
20000610	163	58	51
20000611	150	58	49
20000612	138	60	51
20000613	156	62	54
20000614	174	59	52
20000615	180	55	48
20000616	182	55	47

20000617	159	55	47
20000618	146	56	48
20000619	158	58	49
20000620	167	60	51
20000621	181	61	53
20000622	187	62	54
20000623	165	64	56
20000624	157	66	58
20000625	171	67	60
20000626	168	69	61
20000627	163	70	62
20000628	157	71	62
20000629	167	71	61
20000630	162	68	59
20000701	153	67	58
20000702	151	67	60
20000703	153	68	61
20000704	157	68	60
20000705	156	67	58
20000706	147	67	59
20000707	150	69	61
20000708	152	70	61
20000709	165	70	62
20000710	174	70	62
20000711	157	68	61
20000712	148	68	59
20000713	153	67	58
20000714	165	67	58
20000715	166	69	59
20000716	158	70	60
20000717	159	70	62
20000718	163	71	63
20000719	155	72	63
20000720	157	72	64
20000721	157	73	64
20000722	142	74	64
20000723	135	75	65
20000724	127	75	66
20000725	129	74	66
20000726	131	73	65
20000727	132	74	65
20000728	129	75	66
20000729	129	75	66
20000730	127	75	66
20000731	138	75	66
20000801	143	75	66
20000802	147	73	64
20000803	146	73	64

20000804	153	74	65
20000805	159	75	66
20000806	169	76	67
20000807	175	74	65
20000808	167	72	63
20000809	169	72	63
20000810	165	73	64
20000811	147	74	65
20000812	138	75	66
20000813	134	75	66
20000814	139	76	67
20000815	149	76	68
20000816	166	75	68
20000817	162	76	67
20000818	156	76	67
20000819	149	75	67
20000820	143	74	66
20000821	129	73	65
20000822	129	73	65
20000823	144	74	66
20000824	132	74	67
20000825	134	75	67
20000826	136	76	67
20000827	135	76	67
20000828	134	76	67
20000829	134	75	66
20000830	132	74	65
20000831	137	75	66
20000901	125	75	67
20000902	122	74	66
20000903	119	72	63
20000904	125	71	61
20000905	116	71	61
20000906	121	71	61
20000907	136	73	63
20000908	144	74	64
20000909	136	74	65
20000910	135	73	64
20000911	144	74	65
20000912	154	74	66
20000913	140	74	66
20000914	148	73	65
20000915	176	73	64
20000916	160	72	64
20000917	150	71	64
20000918	147	72	64
20000919	155	73	65
20000920	177	72	65

20000921	163	72	64
20000922	148	72	64
20000923	157	72	63
20000924	167	71	63
20000925	163	70	63
20000926	158	69	62
20000927	137	70	62
20000928	127	71	63
20000929	131	71	63
20000930	140	72	63
20001001	141	72	63
20001002	148	72	63
20001003	143	71	62
20001004	132	70	62
20001005	143	70	62
20001006	177	70	63
20001007	181	70	63
20001008	181	70	62
20001009	174	68	60
20001010	172	66	58
20001011	151	68	60
20001012	137	68	62
20001013	144	68	61
20001014	146	67	61
20001015	126	67	61
20001016	110	67	61
20001017	117	67	61
20001018	140	66	60
20001019	139	65	58
20001020	136	64	57
20001021	134	63	55
20001022	120	62	54
20001023	110	62	55
20001024	115	62	56
20001025	112	63	57
20001026	121	63	57
20001027	130	63	57
20001028	144	64	58
20001029	149	40	36
20001030	147	39	36
20001031	143	41	38
20001101	149	42	39
20001102	139	45	41
20001103	132	45	41
20001104	145	46	41
20001105	164	50	45
20001106	173	52	49
20001107	154	52	50

20001108	156	48	46
20001109	170	45	43
20001110	174	45	41
20001111	157	45	41
20001112	151	43	40
20001113	168	41	38
20001114	185	41	37
20001115	179	42	38
20001116	176	45	40
20001117	196	47	42
20001118	162	46	42
20001119	164	44	40
20001120	185	42	37
20001121	189	41	37
20001122	163	38	35
20001123	135	37	33
20001124	137	35	31
20001125	143	36	32
20001126	167	41	36
20001127	175	39	39
20001128	158	34	32
20001129	159	34	34
20001130	166	30	31
20001201	131	28	30
20001202	132	28	31
20001203	132	30	30
20001204	154	33	32
20001205	177	31	33
20001206	177	32	31
20001207	167	35	34
20001208	140	36	35
20001209	132	38	37
20001210	141	38	38
20001211	156	39	38
20001212	178	40	38
20001213	122	38	37
20001214	153	38	37
20001215	146	41	38
20001216	177	40	37
20001217	205	38	36
20001218	194	35	33
20001219	186	34	34
20001220	185	32	32
20001221	171	29	31
20001222	166	28	29
20001223	142	30	29
20001224	147	34	31
20001225	140	36	34

20001226	151	39	37
20001227	170	40	38
20001228	168	37	38
20001229	173	28	37
20001230	179	30	36
20001231	174	34	33