Panagiotis Leliopoulos

Big Data from Scratch

Building a 4-nodes Hadoop cluster and use of the Map-Reduce Simple Skyline Algorithm (MR-SSA) based on the R Language

About Author

Panagiotis Leliopoulos is holding a Bachelor Degree on Electronic Engineering and three Master Degrees (M.Sc.): On ICT with direction on the Network-Oriented Information Systems and Cloud Computing from the University of Piraeus, on Business Analytics from the Athens University of Economics and Business and on Data Science from the National Centre for Scientific Research "Demokritos", and the University of the Peloponnese.

Hence, he has experience on the Hadoop framework and Map-Reduce processes and he is developing Map-Reduce Skyline Algorithms for Research. Furthermore, he has published articles on e-Commerce, about “The Evolution of Business to Consumer (B2C) E-Commerce” (posted in dblp), and on Education, about “The Use of Big Data in Education” in international conferences and journals.

At the end, he has work experience as Consultant and Technical Project Manager on ICT applications, in the Public and Private sector and Telecommunication Companies.

Abstract

Big Data has been entered into our lives for real. During this book, we make an introduction and analysis on the Big Data. Thus we study the Big Data theory, giving some definitions what Big Data is. Also, we continue analyzing the types of Data and we study the goals of Big Data and see some analyzing techniques. Furthermore, we study some descriptions of previous architectures and systems of cluster computing, like Oscar, Rocks, OpenMosix, and also the root of Map-Reduce, the MPI. Finally, we make a brief reference to the use of Map-Reduce and an introduction to Hadoop framework.

In the other part of this book, we study the performance of a Computer Cluster, developing a Simple Skyline Algorithm (MR-SSA). Moreover, the actual algorithm is designed to run parallel in distributed systems, with the method of Map-Reduce. Specifically, the experimental part of the book consists of a Computer Cluster with four nodes. The development of the algorithm is based on the method of Map-Reduce and implemented within the R language. The fundamental architectural design is including three basic functions like the random range generator, the classification of the dataset in ascending order by column, and at last the calculation of the skyline points. Finally, our application has been tested locally on a computer by using Virtual Machines, which are based on the Cloudera platform.

Table of Contents

1. Introduction

In this Book, we make an introduction and analysis on the Big Data. We begin with a fundamental problem description, what is the big amount of data, where it comes from, how much is the size of the data and how we can process them with the new computing methods and tools we have available. Defining Big Data, we can say “Big Data are a big amount of raw data where does not have any use.”. In this study, we examine the Big Data aspects, where can be utilized from industry and can bring profits.

Hence we study the Big Data theory, giving some definitions what Big Data is. Also, we continue analyzing the types of Data (ex. Big Data, Open Data, Open Government Data). Furthermore, we study the goals of Big Data and see some analyzing techniques, such as Cluster analysis, Crowdsourcing, Data mining, Genetic algorithms, Machine learning, Natural language processing, Network analysis, Data mining, Sentiment analysis and Signal processing. Of course, it is precious to mention, about the purposes of Big Data and where can find applications, for example in Science, Healthcare, Medicine, Real-Estate, Businesses, and Education.

In Chapter 4 we have an extended study of the methods of cluster computing and Map Reduce. Of course, the main issue is to analyze, the vast amount of data according to take results, with an efficient and low-cost way. In this case, we study some clustering methods and conclude the best ways.

Also in Chapter 4, we have some descriptions of prior architectures and systems of high-performance cluster computing, like Oscar, Rocks, OpenMosix, and the ancestor of Map-Reduce, the MPI. We continue to make a more deep study about the architecture and the programming model of the Map-Reduce. Hence we review the Map-Reduce process from the beginning, and we resulting about the methodology of Map Reduce such as the programming model, the architecture and the roles of Map-Reduce. Finally, we make a short reference to the use of Map-Reduce and a brief introduction to Hadoop. Also at this point, we examine all the stages of the method, and we continue with the study of the Hadoop framework, such as its benefits and its components.

In Chapter 5 we analyze what the Hadoop is and what it does, why and in what purpose we use Hadoop. Also, we refer to Hadoop components such as the Hadoop Distributed File System (HDFS), NoSQL, Hive, Pig, Mahout and other expansions for Hadoop programming capabilities, such as Cascading and HBase.

In Chapter 6 we have a look at the vendors and see whose are basing on Hadoop technologies and why they based on Hadoop Framework. After that, it is very fundamental to make an extended reference to Hadoop frameworks. We are referring on the two of the most popular frameworks, Cloudera and Hortonworks.

For the Cloudera Framework, we study about the benefits which provide and see the Cloudera Standard, version 4.8.5 which is a Hadoop Framework setup up in a Virtual Machine. In this VM we are doing our experiments on it, and the reasons for choosing the Virtual Machine method is because it is swift, safe, flexible and we can easily restore that at any time and with excellent results. Of course, the VM method is only for testing reasons, and it is not recommended for commercial use. In the end, we have a look at the Cloudera Data Platform Components like RHadoop.

In Chapter 7 we make an extended study about the Map-Reduce Simple Skyline Algorithm (MR-SSR). Also, we refer to other Map-Reduce algorithms like MR – GPMRS, MR-BNL, MR-SFS, and MR-Bitmap. Furthermore, we are making a study of the structure and implementation of the MR-SSR, and we refer the logic diagram of the algorithm. At the end, we analyze step by step, the pseudocode structure of the MR-SSR.

In Chapter 8 we proceed to our experimental part. We set up and implement the MR-SSR algorithm on a 4-nodes computing cluster, and we proceed taking time measurements in seconds for five different Datasets. After that, we make comments and conclusions about the results of our measurements.

Finally, in Chapter 9, we reach our conclusion for this book, and we are proposing further additional uses for our algorithm.

Hence, in the Annex Chapter, we are seeing an analytical tutorial how to install and run our Map-Reduce Simple Skyline algorithm (MR-SSR) in a 4-node cluster based on Cloudera Framework with Linux Centos operating system, on the VirtualBox which is running on the Windows 10 operating system.

2. Problem Description

2.1 Big Mount of Data that is not using

As we can see nowadays, the Internet has shortened the distances among users all over the world. Those have, as a result, the Internet influences many fields, not only in technical aspects but also to have a high social impact. So as we can see around us, this has application to all kind of electronic services. Furthermore, according to Drigas and Leliopoulos “The Internet is an extraordinary platform for innovation, economic growth and social communication” [1]. Of course, all this traffic leads to the proper growth of bandwidth and data, which generates new needs for new methods of data processing.

Also, the new ways of information development, have been established by the Web. Specifically, the high growth of information is affected by the Web research. Moreover, all of those impacted by the users, whom they use the services of the Web, such as search engines like Google, Yahoo, etcetera. [2].

According to a report, the quantity of digital content on the net is currently near 5 hundred billion gigabytes (for the Year 2010). This amount is anticipated to double within a year. 15 years before, one computer memory unit of information sounded like an enormous amount of data. Now, we tend to hear of information held on in Petabytes unremarkably. Some even verbalize Exabyte or the Yottabyte, that could be a trillion terabytes or, the United website describes it, “everything that there's.” [2].

As A. S. Drigas and P. Leliopoulos mention “from the Year 2012, about 2.5 Exabytes of data are produced each day”, as we can see nowadays the amount of data are doubling almost every 3 years. The result is for everyone second of storing data on the Internet is equal to the whole data of the Internet 2 decades before. Those give businesses a possibility to work with many Petabytes of evidence in a single dataset and not just from the web. For example, it is approximated that Walmart collects more than 2.5 petabytes of data every hour from its customer transactions. According to have in mind how much is a Petabyte (PB), is 1,000 of Terabytes (TB) or 1,000,000 of Gigabytes (GB), or almost 20,000,000 filing cabinets’ price of text. Associate degree Exabyte is 1,000 times that volume or 1,000,000,000 of Gigabytes[2].

2.2 New Computing Methods and Tools

The Computing Technology in the last 20 years has fast grown replacing all the traditional methods. One of the missions of Computing, after all, is to shares content. More business and government agencies are discovering the strategic uses of large databases. New software tools and techniques are considered to analyze the data for better results. A brand new type of "knowledge infrastructure" is being created [2].

Hence the recent Computing technology meets new methods and tools such as Cloud Computing. In this case, Cloud Computing developed as a new tool which includes a bunch of applications such as services over the internet and the developing of powerful computer data centers which can get and distribute these demanded services. Furthermore, this is a set of connected servers which produce significant computing processes and virtualized environment that permits the animated distribution of applications immediately [3].

Across an enormous variety of fields, knowledge measure being collected and expand with quick speed. There's important would like for a brand new creation of procedure theories and tools to help humans in taking helpful info (knowledge) from the very spreading

volumes of digital knowledge.Where can find a lot of applied such as education and others [2].

As Drigas and Leliopoulos say, we can also mention, Computer databases are very useful the last decades. Data is not merely a back-office, accounts-settling tool any longer. Moreover, can be used as a real-time decision-making tool. Researchers using modern interaction methods can now teach possibly useful types of information that otherwise are under of a large amount of data [2].

The real benefit of such data lies in the ability of users to choose useful references, identify attractive events and trends, make support decisions, rely on statistical analysis and reasoning and utilize data to realize business, rational or scientific goals. Once the size of information manipulation, exploration and conclusions grow beyond human abilities, people are looking for new methods to make things easier [2].

2.3 The KDD Process

Below we can see a relative process method about the extraction from raw data, to something useful, such as solid knowledge.

The KDD process is related and repeated at Figure 2.1 (the most of the decisions made by the user). Below we can see nine steps of the process:

1. Learning the application domain: Covers the significant prior knowledge and the targets of the application [4].

2. Creating a target dataset: Covers choosing a dataset or concentrate on a bunch of variables or information samples on that discovery is to be archived [4].

3. Data cleaning and preprocessing: Involves basic procedures, such as eliminating noise if suitable. Gathering the needed information to model or explanation for noise, determining on strategies for management missing data areas, and accounting for time structure information and recognized modifications. As well as determining DBMS issues, such as data types, schema, and mapping of lost and unidentified amounts [4].

4. Data reduction and projection: Contains resulting valuable structures to signify the data, depending on the target of the job, and using capacity decrease or alteration approaches to reduce the current amount of variables under consideration or to discover regular demonstrations for the data [4].

5. Choosing the function of data mining: Contains determining the goal of the model resulting from the data mining algorithm (i.e., classification, summarization, clustering, and regression) [4].

6. Choosing the data mining algorithm(s): Contains picking technique(s) were are using for data forms examination. Such as determining which models and restrictions may be suitable. For example models for categorical data are dissimilar from models on vectors over reals. Also corresponding to a specific data mining method with the total standards of the KDD process. For example, the user may be more focused on accepting the model than in its analytical abilities [4].

7. Data mining: Contains examining for designs of attention in a specific, realistic form or a set of such symbols, covering arrangement rules or trees, regression, clustering, sequence modeling, dependency, and line analysis [4].

8. Interpretation: contains inferring the exposed designs and probably returning to any of the prior phases, as well as the potential conception of the removed designs, eliminating terminated or unconnected designs, and explaining the valuable ones into terms clear by users [4].

9. Using discovered knowledge: contains including this information into the presentation structure, taking activities created on the knowledge, or just documenting it and writing it to involve events, as well as examination for and determining possible encounters with earlier believed (or extracted) knowledge [4].

Previous work on KDD dedicated mostly on the data mining step. On the other hand, the other stages are similar if not more significant for the efficient use of KDD in praxis [4].

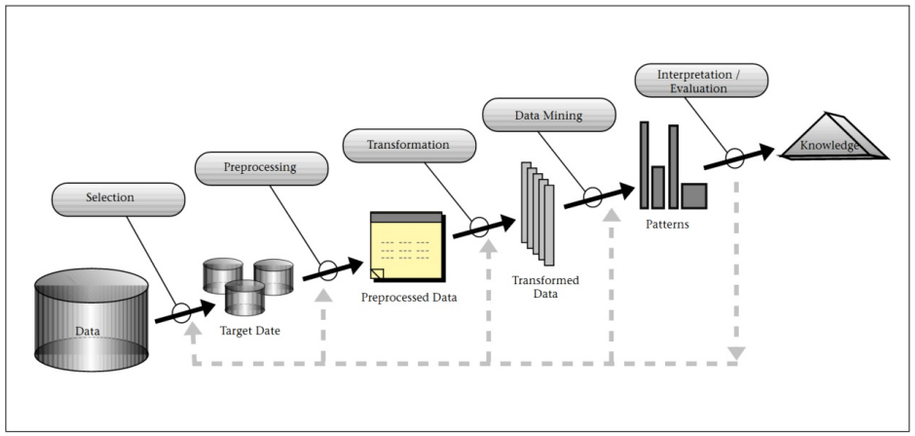

Figure 2.1: Steps Overview of the KDD Process [5].

After all, the final target of this Book is to explain, to analyze and demonstrates the development of a Data Analytics tool, using Hadoop framework, and Map-Reduce method.

3. Big Data Theory

3.1 Definition of Big Data

Well, what is exactly Big Data? We can give some definitions for them. We can say that one aspect of them is the extension of mobile networks, cloud computing and the unknown increase in large amounts of data are defined them as "Big Data" [2].

Furthermore is that the description of progressive techniques and technologies to catch, collection, allocate, accomplish and explore Petabyte or larger-sized knowledge sets with high-speed and varied patterns that foreseeable data management strategies are unable of management [2].

What creates most Big Data; big are various explanations over time and/or area. The online log archives scores of visits daily to a small degree of pages. The cellular phone information stores time and position often, for every mobile user. The vendor has thousands of stores, tens of thousands of product, and uncountable customers, however, logs billions and billions of separate connections during a year [2].

According to the above definitions, Big Data can provide advanced parallel techniques, such as data analyzers (human and computer), mass information to predict situations such as activities and processes in some ways they had not realized years ago [2].

Another aspect, as reported by Drigas and Leliopoulos, "Big Data is datasets whose size is beyond the capability of standard database software tools to capture, store, manage, and analyze." In this case, new technological techniques such as the Hadoop Framework, Map-Reduce methods, and Database Techniques such as data mining to extract knowledge from databases are used [2].

Therefore, Big Data has confirmed the magnitude of estimating growth, saving money, increasing productivity and improving decision-making in areas such as traffic control, weather forecasting, disaster prevention, financing, control fraud. Finally, as we can see from above, they result in many basic changes to the data. Creating a replacement era within which all processes and interactions, together with a research project, are affected [2].

3.2 Big Data and Open Data

As the McKinsey Global Institute says, Big Data refers to datasets that are bulky, diverse and up to date. On the other hand, Open Data is, in fact, Big Data but smaller and the information is open to everyone [2].

As we can see, Open Data comes mainly from Government datasets or other institutions, businesses and individuals. In fact, the public sector is fueling open data, and the public sector includes schools, universities and other educational institutions [2].

In Figure 3.2 we can see the distribution of all world data. We know that Big Data is an important part of them. Also, Big Data included Open Data and then, the Open Government Data. Finally, the concept of Open Data related to "MyData," which involves the exchange of information collected for an individual or organization with that person, such as hospitals [2].

To be able to make an overview, Open Data is an essential part of the Big Data category and has a great impact on them, but also in our Book, as we see below.

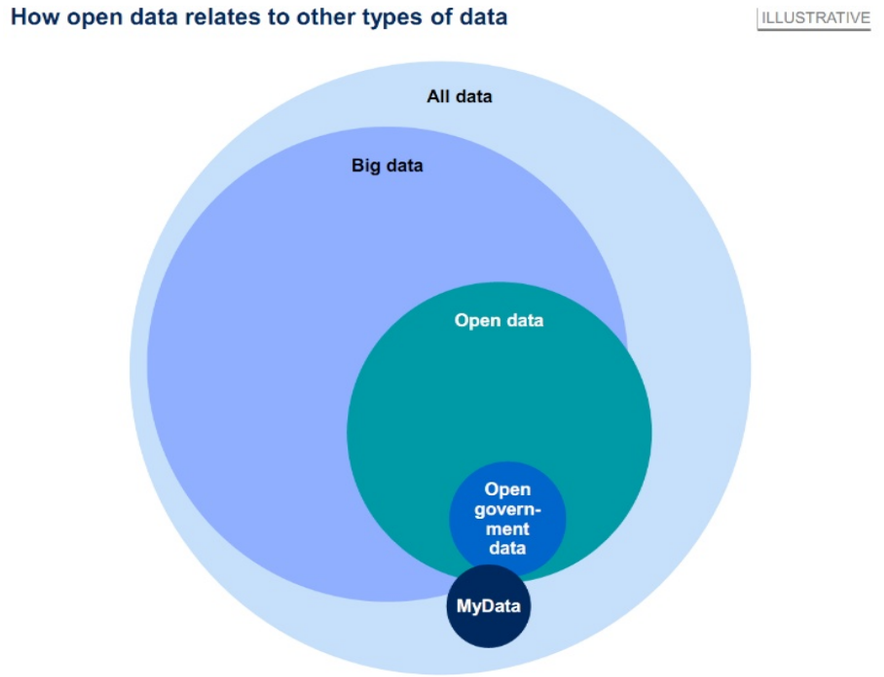

Figure 3.1: How open data relates to other types of data [2].

3.3 Goals of Big Data

As we can see, we are now surpassing a data spill. In an extensive range of application areas, data is collecting on an extraordinary scale. Decisions made earlier on the assessment or well-formed reality models can now be determined in the same data. This massive data analysis is now working on almost every aspect of our current culture, including mobile services, retail, producing, money services, life sciences, physical sciences and education [2].

3.3.1 Techniques for analyzing Big Data

Below we can see some of the techniques for analyzing Big Data and a short description of them.

1. **Cluster analysis**

Cluster analysis we mention the partitioning of data into significant subsections, when some subgroups and other evidence about their configuration may be unidentified [6].

2. **Crowdsourcing**

Crowdsourcing is a newly developed term that states the process of outsourcing actions from a Company to an online community or crowd in an "open call" process. Any partner of the group can then complete a distributed job and be rewarding for their difficult work [7].

3. Data mining
Data mining and knowledge finding in databases have been interesting a major quantity of research, industry, and media attention at least 20 years [5].

4. Genetic algorithms
Genetic Algorithms are powerful search and optimization methods which have used in an amount of applied issues. The strength of Genetic Algorithms is owing to their volume to find the total optimal in a multimodal background. A plenty of such multimodal tasks is in engineering problems such as the development of neural network structure and learning neural network weights, solving optimal control problems, designing structures, and solving flow problems and much more [8].

5. Machine learning
Machine Learning is keen to papers about genetic algorithms and genetics-based learning systems. Just specified, genetic algorithms are probabilistic search actions intended to effort on big spaces including conditions that can be signified by strings [9].

6. Natural language processing (NLP)
Natural Language Processing (NLP) may be a series of analysis and applications that discover however computers are wont to acknowledge and use normal language or speech to form valuable things. Information processing applications embrace a variety of study areas, like mechanical translation, reduction and text redaction, user interfaces, Multilingual and Cross-Language Information Retrieval (CLIR), speech recognition, artificial intelligence and specialized systems [10].

7. Network analysis
Big networks, consuming thousands of peaks and outlines, can be established in numerous altered areas in example genealogies, flow graphs of programs, molecule, computer networks, transportation networks, social networks, intro or inter-organizational networks. Several common network algorithms are actual time and space overwhelming and consequently inappropriate for analysis of such networks [11].

8. Sentiment analysis
The unexpected upsurge in action in the field of opinion mining and sentiment analysis. Which is shrinking with the computational behavior of opinion, emotion, and subjectivity in the text. Sentiment analysis has at least partially been a direct response to the flow of attention to the new systems dealing with views as first-class objects [12].

9. Signal processing
As minimal-error approximation causes willpower, this solid is at the border of signal processing, computational learning, and pattern recognition. Numerous signal-processing issues influence application: the signal model, morphological operator representation, and desirable operator possessions [13].

3.3.2 Analyzing the data
In the last years, we can see a big growth of data analytics. This aspect has reduced the costs of the data. So we can say more and more data created giving the opportunity to companies to collect data that are extremely fine.

The meager budget of data storage is very attractive to the companies about the high-quality results. Those give new aspects of the industries such as to better understand their customers.

Below we can see some special features of the Big Data phenomenon as they relate to analytics:

- Discovering text and semi-structured data to get if these sources could offer an extra vision.
- Tapering the time gap among data achievement and interim on a business decision created on the data, occasionally mentioned to as near real-time business analytics.
- Testing with deep analytics outside the functionality obtainable by the old-style business intelligence (BI) stack.
- Looking for low cost, highly scalable analytics platforms [14].

As we can see the fast and profitable analytics over “Big Data” has appeared as a fundamental achievement component for many companies. Such as for scientific and engineering disciplines, and government endeavors [15].

Web search engines and social networks taking and evaluate every user action on their sites to develop site design, spam and fraud detection, and advertising events. Strong telescopes in astronomy, genome sequences in biology, and particle accelerators in physics are providing significant volumes of data to the scientists. Fundamental scientific progress is okay to come from computational analysis of such data. Several necessary and tested science disciplines now have computational fields, e.g., computational biology, computational economics, and computational journalism [15].

Collection, range, opportunities, involvement and separate problems with Big Data slow the growth at all stages of the pipeline that can create value from data. The issues begin immediately until data addition when the data waves depend upon us make decisions. Directly specific way about what data to keep and what to reject, and how to collect what we keep probably with the right metadata [16].

Plenty data today is not naturally in a structured format, for instance, tweets and blogs are weakly structured parts of the text, while images and video are structured for storage and presentation. But not for semantic content and search, the reconstruction such content into a structured format for after analysis is a significant demand. The profit of data blasts when it can be connected with other data, thus data integration is a major creator of value [16].

Since most data is straight created in digital format nowadays, we have the convenience and the demand together to control the production to promote later correlation and to connect previously created knowledge commonly. Knowledge analysis, organization, retrieval, and modeling, measure alternative basic challenges. Data analysis is a little obstacle in plenty functions. Two together due to the need of scalability of the core algorithms and due to the involvement of the data that is necessary to be analyzed. Certainly, demonstration of the

events and its analysis by non-technical authority experts is vital to obtain actionable knowledge [16].

3.4 Purposes of Big Data

If we want to ask ourselves, what are the purposes of "Big data" for business, we can say, is the analyzing of every transaction. Big Data is the capturing insights from every customer interaction. Without to wait for months to get the data from the participants [17]. Furthermore, the data are growing twice every 18 months and as a result from to customer data from public, proprietary, and purchased sources. Hence new information gathered from Web communities and newly deployed smart assets [17].

As we can see, these directions are commonly known as "Big Data." Technology for taking and analyzing information is commonly accessible at ever-lower price points. On the other hand, many businesses are taking data use to new standards, using IT to support proper, stable business experimentation that guides choices and to test new products, business models, and innovations in customer experience. In some instances, new ways help companies make decisions in real time. This thing can effort a vital revolution in research, innovation, and marketing [17].

At the same time, the number of data that needs to be stored and processed by analytical database systems is growing. These are partially due to the expanded automation with which data can be composed, as more companies go to a digital era, the expansion of sensors and data-producing devices, Web-scale communication with customers, and government consent demands along with strategic firms drive to demanding more historical data be kept online for analysis [18].

It is no more extraordinary to hear of businesses to collect to load over a Terabyte of structured knowledge per day into their analytical info system and keeping data warehouses of size more than a Petabyte [18]. Also taking a look at the market, we can see, the analytical database market directly rests of $3.98 billion of the $14.6 billion database software market (27%) and is developing at a rate of 10.3% per year [18].

Venture capitalists are very much familiar with this thing and have funded many businesses in last years that build specializing in analytical information management code and continue to fund them, even in hard economic days [18]. The conclusion from all of them is a new epoch of Big Data is rising, and the meaning of business, government, democracy, and culture are huge [19].

3.4.1 The multiple roles and use of Big Data

Here we can mention some of the multiple functions and use of Big Data.

1. Science

Scientific research has been reform from Big Data. The digital data analysis has been nowadays the main source for astronomers all over the world. The area of Astronomy is regenerating from simply taking photos from the sky, where the photographs are already stored information, and therefore the astronomer's task is to find spectacular objects and phenomena within this information [20].

In the biological sciences, there is now an accepted attitude of collecting scientific data into a public repository and also of building public databases for use by other scientists. In case

there is a full specialty of bioinformatics that is widely dedicated to the duration and analysis of such data. As technology progress, especially with the advent of Next Generation Sequencing, the amount and the quantity of the experimental datasets accessible is developing expanding [20].

2. Healthcare

It is accepted that the use of information technology can lower the price of healthcare while bettering its condition. Creating responsibility protective and illustrated and depended it on more expanded repeated controlling. It is believed a savings of 300 billion dollars every year in the US alone [16].

Behind this stream of inspiration, plenty vendors are present healthcare solutions and services such as telemedicine, electronic medical records, medical imaging, and patient management that can be expanded or combined by healthcare providers, clients, and consumers over a cloud. For instance of such an application in health care is the famous Microsoft Health Vault, it is a web-based platform from Microsoft used to store and keep health and fitted data [21].

3. Medicine

Also as we can see Big Data can found the application and uses such as in medicine, where computational physics and various other disciplines have to deal with large volumetric data sets that demand an adequate visualization [22].

4. Real-Estate

The real-estate business, for example, deals with on information asymmetries such as great access to purchase data and hard held knowledge of the offer and ask action of customers. Together require considerable expense and effort to achieve [23].

5. Businesses

Big data leaders in the opportunity of a basically another type of decision-making. They are using composed analysis, businesses can test hypotheses and evaluate the results to guide assets decisions and useful trades [23]. Customer-facing industries have long used data to piece and destination customers. Big data accepts a large progress above what until lately was studied updating, by building potential real-time examples [23].

Big data increases the useful field of algorithms and machine moderate analysis. At some corporations, for instance, algorithms analyze sensor data from manufacturing lines, creating self-adjusting cases that cut waste, escaping price and sometimes serious human interruptions, and amount lifting [23]. Big data is increasing the new group of businesses that grasps data-driven business types. Plenty of these firms plays connection roles in value chains where they catch themselves generating helpful "exhaust data" created by business activities [23].

Business intelligence, Business Analytics (BI&A) and the relevant area of Big Data Analytics has become more and more crucial in both the academic and the professional associations over the past two decades. Business studies have emphasized this vital development [24].

Established on a sample of over four thousand information technology (IT) professionals from ninety-three countries and twenty-five industries, the IBM Tech Trends Report (2011) classified business analytics as one of the four serious technology directions in the 2010s. In a study of the state of business analytics by Bloomberg Businessweek (2011), 97% of

industries with top credits $100 million were discovered to use some form of business analytics [24]. It is predicted that by 2018, the United States alone, probably face a lack of 140,000 to 190,000 people with excellent analytical skills, as well as a loss of 1.5 million data-smart managers with good knowledge to analyze big data and create active opinions [24].

6. Education

The use of ICT has been implemented in most activities of modern applications. In this framework, it could not accept education as the needs of new technologies has been increased in recent years. In particular, the turning in the use of ICT in education has often driven, a more education-oriented student approaches [25].

Big Data has the longer term to alter not solely analysis however conjointly education. A late correct vital similarity of the many approaches taken by thirty-five charter faculties in NYC has discovered that one amongst the highest five policies connected with important tutorial effects was the employment of knowledge to guide instruction. Different collaboration Technologies that the massive knowledge ar supported them is that the Cloud Computing. These technologies are improving the academic services, giving young and adult students alike access to inexpensive subjects, online instructors, and fellow learners societies [2].

Furthermore, as Drigas and Leliopoulos say “Big Data can support the classic educational system helping teachers to analyze what students know and what techniques are most effective for each pupil.” In this way, also teachers can learn new technologies and methods about their education work [2].

Hence, technologies like data processing and Data Analytics are giving a quick feedback to students and lecturers regarding their educational performance. These ways are giving a deep analysis of some education patterns and extract valuable information from them. During this method, collective and massive scale knowledge are predicting United Nations agency student desires additional facilitate from the education system, avoiding the danger of failure or drop out. All these have, as a result, to search out pedagogical approaches that appear only with explicit students and special needs [2].

On the other hand, as Drigas and Leliopoulos say “Big Data can easily find application in online education”. As we can see, the web education includes a massive development in recent years and includes an increasing impact on the education sector. What is more, digital learning is a set of information and analytics which may contribute to teaching and learning. During this method, several students participate in online or mobile learning, wherever area unit created new information. This new information, additionally with the assistance of social networks, area unit serving to the scholars with the various background to correlate between them and facilitate them to know core course ideas [2].

4. Methods of Cluster Computing and Map Reduce

4.1 High-Performance Cluster Computing: Architectures and Systems

In this chapter, we are seeing a short description of previous high performance and clustering computing technologies, where had taken a crucial part for the future development of the map-reduce method, and Hadoop. Furthermore, we have a look at the Oscar, Rocks, OpenMosix, and the ancestor of Map-Reduce, the MPI. Finally, we study the methods of Map-Reduce extensively.

According to Bilbao *et al*. “Clustering is a technology or a set of technologies that allow multiple computers to work together to solve common computing problems.” So we can mention, the computing problems can be everything from complex CPU-accelerated scientific computations to a crowd of different processes with no basic common [26].

In the beginning, clusters were created to solve issues of super computation, but today it is not their only purpose. The growth of the need for the web technology has caused the application of clusters in various servers with the aim of service to a large number of clients. This category of technology has been performing in services such as web servers, email, e-commerce, or high-performance databases [26].

Hence, cluster computing systems give the storage capacity, computing power, and high-speed local area networks to manipulate large data sets. In association with “new forms of computation combining statistical analysis, optimization and artificial intelligence,” as Bollier *et al*. said, researchers “Can construct statistical models from large collections of data to infer how the system should respond to new data.” [19].

In fact, the cluster is a class of parallel or distributed processing system, which is an amount of connected stand-alone PCs that work together as an individual, integrated computing resource. A computer node may be just one or a multiple processor systems (Personal Computer, Workstation or SMPs) with memory, Input / Output peripherals, and OS [27].

A cluster mostly refers to two or more, connected computers (nodes). The nodes can exist in a single case or be physically apart and linked via a LAN. An inter-connected (LAN-based) cluster of computers can be developed as a single system to users and applications. Such a system is affording economic, thanks to accomplishing options and edges (fast and reliable services) that have traditionally been found solely on new pricey recovery shared memory systems. The typical architecture of a cluster is shown in Figure 4.1. [27].

Below we can examine some relevant cluster technologies.

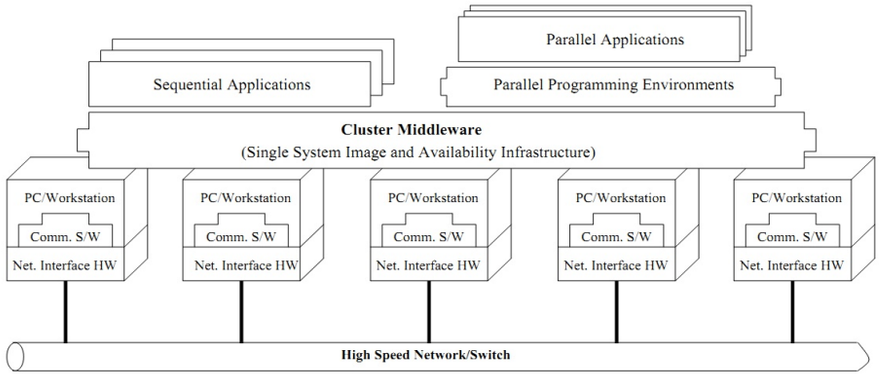

Figure 4.1: Cluster computer architecture [27].

4.1.1 OSCAR

Open Source Cluster Application Resources (OSCAR) is an entirely united package of software planned for the building, maintaining, and using a Linux cluster for high-performance computing (HPC). As the first overview of OSCAR, this software package has developed a well-accepted select for constructing HPC clusters. As cluster computing systems still are applied to mission-critical situations, high availability (HA) structures are consequently necessary to be involved in OSCAR cluster [28].

The hardware structure of HA-OSCAR cluster is displayed in Figure 4.2 the system contains the main server, a standby server, two LAN connections, and several clients, where all the clients have similar hardware. A server is accountable for serving user's requirements and allocating the needs to particular clients. A client is devoted to computation. Each server has three network interface cards, one is connected to the web by a public network address, and therefore the difference two is connected to a non-public LAN, which contains a primary Ethernet LAN and a standby LAN. Each LAN contains network interface cards, switch, and network wires, and offers communication between servers and clients, a connection between the primary server and therefore the standby server [28].

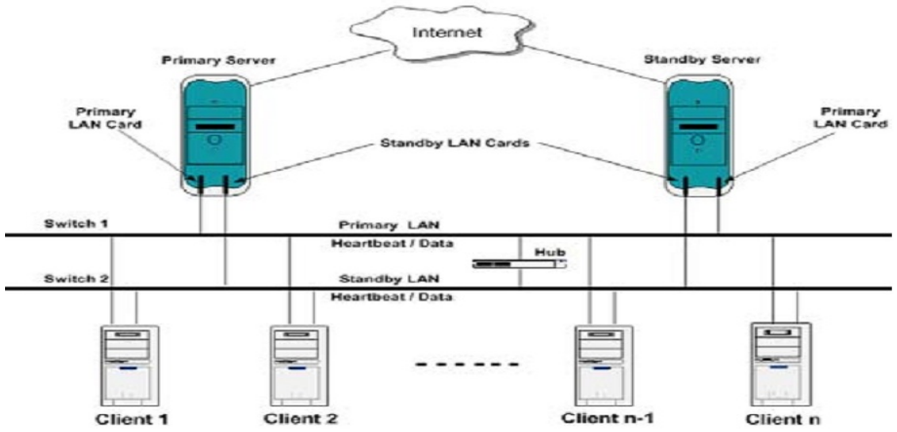

Figure 4.2: Architecture of the HA-OSCAR cluster [28].

4.1.2 Rocks

While prior clustering toolkits use an extensive effort to match configurations of nodes, Rocks creates complete Operating System installation on a node with a basic managing tool. With care to full automation of this procedure, it develops quicker to reinstall all nodes to an identified configuration than it is to control if nodes were out of management in the first place. Dissimilar a user's desktop, the Operating System on a cluster node is careful to be a soft state that can be reformed and/or updated quickly. Those are of course opposite to the viewpoint of configuration management tools like Cfengine that achieve complete analysis and equality testing of an installed Operating System [29].

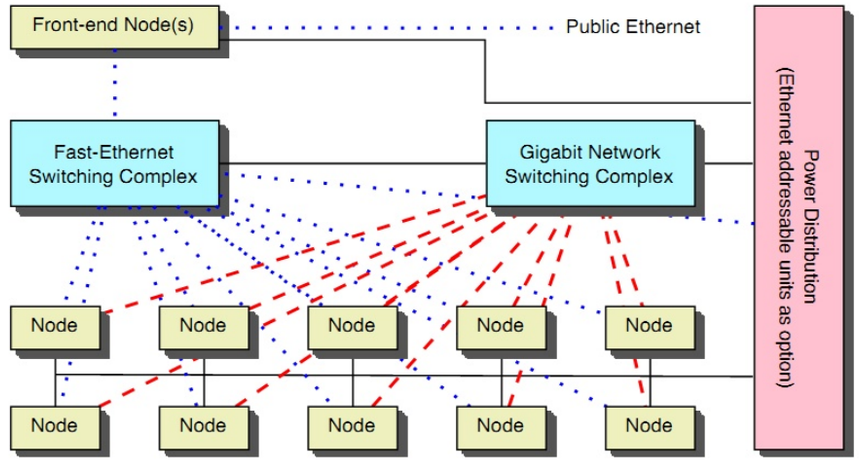

Figure 4.3: Rocks hardware architecture. Based on a minimal traditional cluster architecture [29].

Figure 4.3 shows a traditional design commonly used for high-performance computer clusters set up by the Network of Workstations and promoted by the Beowulf project. This system is harvested from standard large volume, Ethernet, electrical, and optional high-performance cluster connections (e.g., Myrinet or Gigabit Ethernet). We have defined the Rocks cluster architecture to cover a simple set of high-volume modules to figure consistent systems by dipping the module count and by using modules with high mean-time-to-failure specifications [29].

4.1.3 OpenMosix

Essentially, OpenMosix is a Linux kernel addition for single-system image clustering. This kernel addition turns a network of usual computers into an ultra-computer for Linux implementations. Once we have setup OpenMosix, the nodes in the cluster start communicate to one another, and the cluster adjusts the situation to the amount of work. Procedures creating from any one node, if that node is too full of activity associated with others, can transfer to any other node [26].

4.1.4 MPI

MPI is an application messenger interface that transmits messages, consisting of protocols and semantic qualifications for how to run its structures in any function. MPI includes the transmission of point-to-point messages and common functions, all of which aim at a set of user-defined procedures. Also, MPI provides concepts for processes on two levels [30].

- Procedures are called according to the rampant of the group in which the communication is being accomplished.
- Simulated topologies permit for a graph or Cartesian naming of procedures that service share the application semantics to the message passing semantics in a suitable, effective technique [30].

Correspondents, which house groups and message setting information, offer a significant amount of protection that is needed and beneficial for constructing a library-oriented parallel code. MPI also provides three extra classes of services: conservational analysis, elementary timing data for application presentation amount, and a summarizing interface for outside presentation monitoring. MPI creates various data change a light quantity of its services by needing data type description for all communication procedures. Both built-in and user-defined data styles are providing [30].

4.1.5 The Map-Reduce

As the Internet range retains, its developing, enormous data requirements are handled by several Internet Service Providers. The Map-Reduce framework is now becoming a primary instance result for this. Map-Reduce is intended for configuration big service cluster, which contains thousands of nodes by spending service hardware [31].

The performance of a parallel system like Map-Reduce system carefully links to its task scheduler. Many scientists have presented their attention in the schedule issue. The existing scheduler in Hadoop uses a single queue for scheduling jobs with an FCFS technique. Yahoo's company capacity scheduler as well as Facebook's company appropriate schedule procedures many queues for dealing altered resources in the cluster. Applying this scheduler, users could allocate jobs to queues which could only by hand guarantee their precise resource share (Figure 4.4) [31].

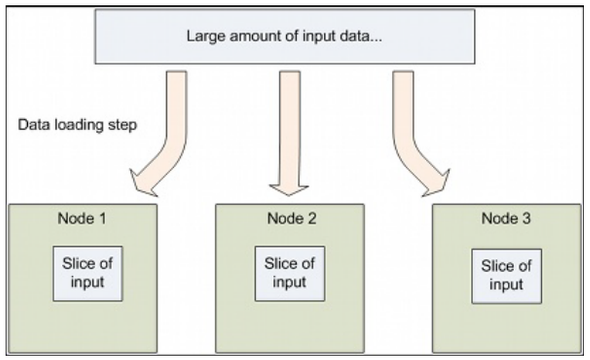

Figure 4.4: Large amount of data separating to Nodes [32].

Map-Reduce is a theoretically remote plain parallel processing system than old HPC systems. Current algorithms, reformed somewhat to work on data as stored on HDFS, frequently run very well. Map-Reduce is capable of performing Java code, but also to procedure software written in other languages, including high-level languages like C or C++ and scripting languages like PHP, Python, and Perl. Hence, an amazingly significant amount of algorithms can be expressed as Map-Reduce procedures [33].

Map-Reduce similarly bests at complete processing. If an algorithm needs to study every single record in a file in command to calculate an effect, then Map-Reduce is a brilliant choice. Jobs run in parallel on Data-Nodes in a cluster. The amount of Data-Nodes in a cluster is straight relative to the quantity of data the cluster can store. The outcome, the size of a dataset affects the performance of a Map-Reduce job fewer than the difficulty of the algorithm it implements (Figure 4.5) [33].

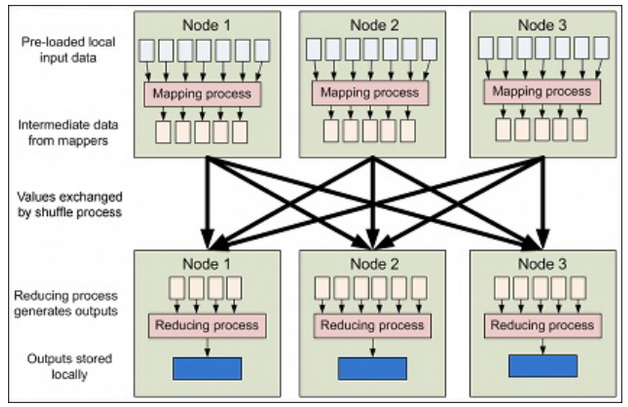

Figure 4.5: Map-Reduce processing to Nodes [32].

Applications written in this efficient method are inevitably parallelized and performed on a big cluster of service machines. The run-time system takes care of the parts of partitioning the data input, scheduling the program's performance through a set of machines, controlling machine disasters, and handling the necessary inter-machine communication. These permits programmers without any skill with parallel and distributed systems to simply use the resources of a big distributed system [34].

Map-Reduce is a highly scalable method and can be utilized through several computer nodes. Furthermore, Map-Reduce is appropriate when the object issue is accurate data concentrated, and the computing setting has limits on multiprocessing and big shared-memory machines. Map-Reduce transfer the processing to the data and procedures data in sequence to evade random admission that needs costly searches and disk quantity. Map-Reduce technologies have furthermore been accepted by a developing amount of sets in the industry (e.g., Facebook, and Yahoo). In the academic world, scientists are discovering the usage of these examples for scientific computing, such as in the ranges of Bioinformatics and Geosciences where codes are written using open source Map-Reduce tools [35].

4.2 About the Methodology of Map Reduce

4.2.1 Programming Model

The Map-Reduce programming model, presented by Google, has come to be a standard as a different framework for parallel data programming over the previous few years related to the message passing methodology [35]. Furthermore, the Map-Reduce programming model is driven by functional languages and targets data-intensive computations. The input data format is application-specific and is specified by the user [36].

Big Data clustering with Map-Reduce is a quick result to frame the clustering algorithm that completes amazing value effects within an adequate quantity of time. Besides, the parallelization with Map-Reduce is amazing for the reason that it grants a programming model that parallelizes the jobs automatically, and offers fault-tolerance and load balancing. [35].

As we can see, the fundamental idea back of the Map-Reduce approach is that the issue is expressed as a clever concept spending two core procedures: the Map procedure and the Reduce procedure. The Map procedure repeats over a big amount of accounts and excerpts remarkable data from each record and all standards with the similar key are directed to the similar Reduce procedure. Moreover, the Reduce procedure totals middle consequences with the similar key that is produced from the Map procedure and then creates the final events. Figure 4.6 shows the Map-Reduce's core procedures [35].

Map Procedure:
Map (k, v) → [(k', v')]

Reduce Procedure:
Reduce (k', [v']) → [(k', v')]

Figure 4.6: The Map and Reduce Procedures [35].

A more detailed description is:
The result is a set of <key, value> set. The user declares an algorithm using two jobs, Map and Reduce. The Map job is applied on the input data and exports a list of middle <key, value> pairs. The Reduce function is implemented to all middle sets with the similar key [36].

It characteristically achieves approximately a kind of integration procedure and exports zero or more output sets. Lastly, the output sets are sorted by their key value. In the simplest form of Map-Reduce programs, the programmer offers just the Map job. All other functionality, as well as the combination of the middle sets which have the similar key and the final categorization, is providing by the runtime [36].

The next pseudo code displays the simple construction of a Map-Reduce application that amounts the quantity of incidences of for each word in a group of documents. The map job emits each word in the documents with the provisional count 1. The reduce job sums the counts for every single word [36].


```
// input: a document
// intermediate output: key=word; value=1
Map(void *input) {
for each word w in input
EmitIntermediate(w, 1);
}
// intermediate output: key=word; value=1
// output: key=word; value=occurences
Reduce(String key, Iterator values) {
int result = 0;
for each v in values
result += v;
Emit(w, result);
}
```
[36].

The main benefit of this method is easiness. The developer offers a simple explanation of the algorithm that emphasis on performance and not on parallelization. The real parallelization and the particulars of concurrency organization are left to the runtime system. Therefore the program code is basic and simply manageable across systems. However, the method offers sufficient high-level data for parallelization. The Map job can be performed in parallel on non-covering shares of the input data, and the Reduce job can be performed in parallel on each set of middle sets with the similar key [36].

4.2.2 Architecture

The P2P-Map-Reduce design contains three simple parts, shown in Figure 4.7. : User (U), Master (M) and Slave (S). Master nodes and slave nodes form two logical peer to peer networks called M-net and S-net, separately. As stated above, computing nodes are animatedly allocated the master or slave role. Also, M-net and S-Net alternate their arrangement over time [37]. Also, we can see, through an example, how a master failure is controlled in the P2P-Map-Reduce design. We accept the starting condition characterized in Figure 4.7. , where U is the user node that acquiesces a Map-Reduce procedure, nodes M are the masters and nodes S are the slaves [37].

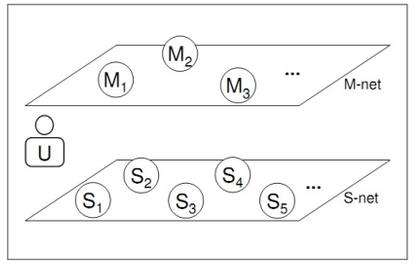

Figure 4.7: P2P Map-Reduce Design [37].

The next stages are executed to resign the job and improve from a master failure (see Figure 4.8.):

1) U queries M-net to get the list of the existing masters, each one described by a job index that processes how busy the node is. U orders the list by mounting values of job index, takes the first component as primary master and the next k elements as backup masters. For example, M1 is selected as the primary master and M2 and M3 as backup masters (k = 2). Then, U succumbs the Map-Reduce job to M1 along with the names of its backup nodes (M2 and M3) [37].

2) M1 notifies M2 and M3 that they will act as backup nodes for the existing task (in Figure 4.8., the apex "B" to node M2 and M3 shows the backup task). These suggest that M1 will occasionally backup the whole task state (e.g., the jobs of tasks to nodes, the places of intermediate results, etc.) on M2 and M3, which in opportunity will occasionally check whether M1 is active [37].

3) M1 queries S-net to get the list of the existing slave, selecting (part of) them to complete a map or a reduce job. As for the masters, the select of the slave nodes to use can be done on the source of a job index and other concert metrics (e.g., CPU speed). In this sample, nodes S1, S3 and S4 are designated as slave nodes. The jobs are in progress on the slave nodes and achieved as normal in Map-Reduce [37].

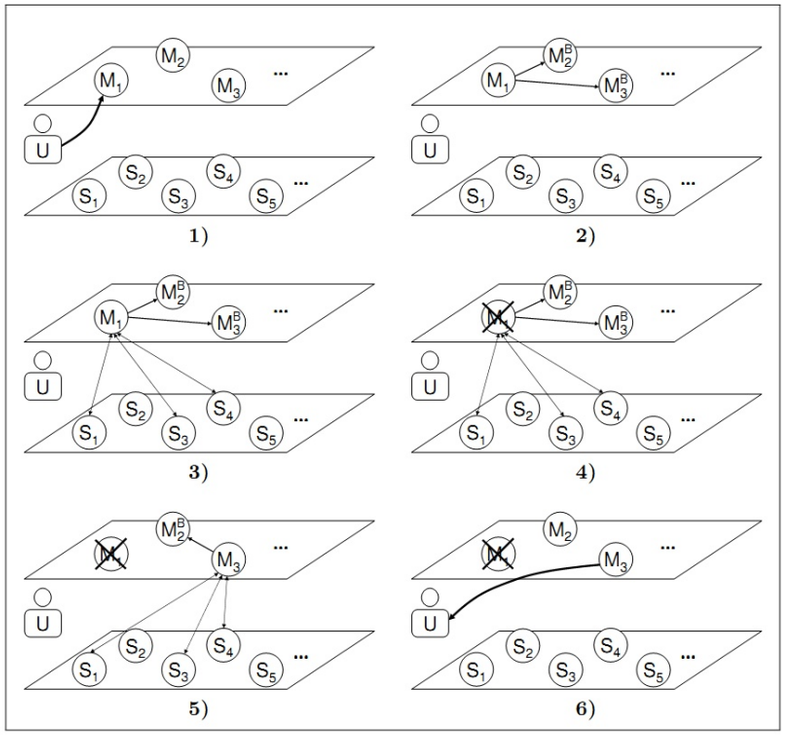

Figure 4.8: Steps performed to submit a job and manage a master failure [37].

4) The primary master M1 fails. Backup masters M2 and M3 notice the failure of M1 and start a spreading process to choose between them the new primary master [37].

5) The new primary master (M2) is selected by selecting the backup node with the lowermost job index. The residual k ¡ 1 backup nodes (only M3, in this example) remain to play the backup master role. Then, the Map-Reduce job resumes from the most recent checkpoint existing on M2 [37].

6) As the Map-Reduce job is complete to the M2, precede the result of the U. It is crucial to notice that the master failure and the recovery job are obvious to the user. It must also be well-known that a master node may play at the same time the role of primary master for one task and that of the backup master for another task. Figure 4.9 shows an instance of such a condition, in which:

- User nodes U1 and U2 have resigned their Map-Reduce tasks (correspondingly Job 1 and Job 2).
- M1 is the primary master of Job 1, and its backup masters are M2 and M3.
- M3 is the main master of Job 2, and its backup master is M4 (also, M3 is at the same time backup master for Job 1 and primary master for Job 2) [37].

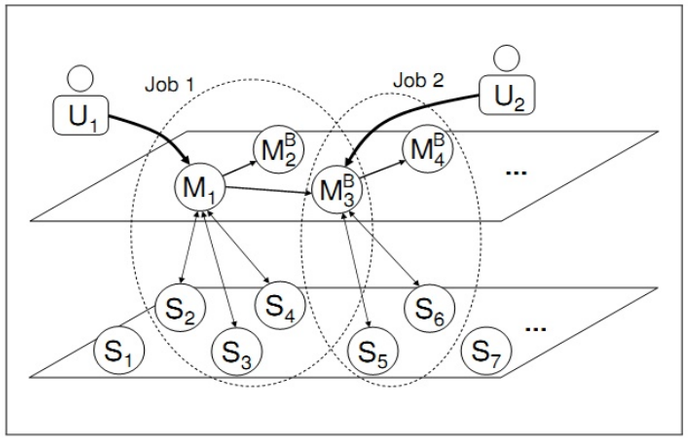

Figure 4.9: Master nodes are playing multiple roles for different jobs [37].

4.3 Roles of Map-Reduce

4.3.1 Map-Reduce computations with CGL-Map-Reduce

The altered phases, which the CGL-Map-Reduce permits complete throughout usual Map-Reduce calculations, are displayed in Figure 4.10, and the depiction of each phase follows.

1. Initialization Stage

The first phase in using CGL-Map-Reduce is to start the Map-Reduce job-workers and design together with the map and reduce jobs. CGL-Map-Reduce cares designing map/reduce jobs and reprocessing them many times with the purpose of supportive constant Map-Reduce calculations professionally. The job-workers collective the designed map/reduce jobs and use them when an application is established from the user to complete the map job. This structure phase, which happens only after, can be recycled to load any permanent data needed for the map jobs. For usual single pass Map-Reduce calculations, the user may not want to tool the pattern phase [38].

2. Map Stage

The second phase after the job-workers are reset, the user application initiates the MR-Driver to start the map calculations by fleeting the mutable data (<key, value> pairs) to the map jobs. MR-Driver transmits this to the job-workers, which then appealed the designed map jobs. This method permits the user application to license the effects from a prior repetition to the following duplication in the case of iterative Map-Reduce. The exports of the map jobs are moved straight to the suitably reduce job-workers with the content distribution network [38].

3. Reduce Stage

The third phase, reduce job-workers are resetting in the similar method as the map job-workers. Then reset, the reduce job-workers delay for the map exports. MR-Driver educates the reduce job-workers to begin performing the reduce jobs after all the map jobs are accomplished. The output of the decrease calculations is also directed straight to the user application [38].

4. Combine Stage

At the fourth phase, when the user application accepts all the exports of the reduce calculations; it may achieve a chain process stated by the user. For a standard single pass Map-Reduce calculation, this phase can be used to chain the outcomes of the reduce jobs to produce the last effects. In the occasion of iterative Map-Reduce calculations, this phase can be used to calculate the critical value to ensure the repetitions [38].

5. Termination Stage

At the final fifth phase, the user application updates the MR-Driver its rank of finishing the Map-Reduce calculation. The MR-Driver also ends the set of job-workers used for the Map-Reduce calculation [38].

4.3.2 Map-Reduce procedure analysis

The Map-Reduce encloses a map stage group data indefinite key, and a reduce stage combining data scuffled from map nodes. Map jobs are a container of autonomous jobs which use altered input. They are allocated to different nodes in the cluster. Hence, reduce jobs depend on the export of map jobs. They have becoming map outcome data from other nodes. Scuffle activities which are I/O-bound frequently intercross in the map stage, for making the most of the value of I/O source. As shown in Figure 4.10, after all, wanted middle data to get scuffled, the reducer creates to calculate [31].

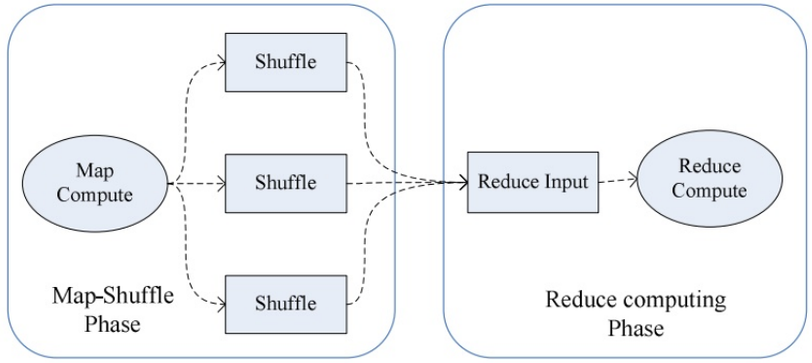

Figure 4.10: The Map-Reduce data process stages [31].

We decay the Map-Reduce method into two sub-stages which are a Map-Shuffle stage and Reduce-Computing stage. In the Map-Shuffle stage, every node amount five movements:

1. In it input data.
2. Calculate map task.
3. Supply output outcome to local disk.
4. Scuffle the map tasks outcome data out.
5. Moreover, scuffles reduce the input data. The Map-Shuffle stage is the first stage. Moreover, in the Reduce-Calculating stage, jobs could straight begin to run the application logic as the data of entry is at present scuffle in memory or local disk [31].

In this assessment of Map-Reduce, the Map-Shuffle stage is serious to the total process. In this stage, each node executes their map job logic which is comparable in one task, and scuffles affect data to all reducer nodes. All reducer nodes are not capable of activating calculating for just one map node slows down [31].

4.4 Use of Map-Reduce

While examining a variety of applications for which the Map-Reduce can be particularly operative, we observed that by backup iterative Map-Reduce calculations we could increase its applicability to additional areas such as machine learning, computer vision, and data clustering, where several iterative algorithms are the same. In these algorithms, Map-Reduce is used to lever the parallelism while the repetitive repeated of it finalizes the repetitions [39]. Furthermore one of the benefits of Map-Reduce comparative to parallel databases is cost. There happens an open-source version of Map-Reduce (Hadoop) that can be found and used for free [18].

4.4.1 From Map-Reduce to Hadoop

Map-Reduce and Hadoop were made when Google and Yahoo! started resolving an engineering problem core to their industries: creating a guide of further than one billion web pages. The existing mythologies have confirmed valuable there. However, it has established even more important in other, unanticipated areas like developing page design, advertisement collection, spell checking, map rendering, and so on. An overall purpose tool that grand analysis of all data allows new studies that weren't before applied. Providing to all developers easy, powerful admissions to all data produces an astonishing crowd of business development [33].

Additional data in and of itself creates excellent value information. If an individual can admission to the data, designs appear that are not obvious with fewer data. In a world where information divides leaders from failures, successful organizations will decide themselves by how they work with data. Their creators define some brilliant cases in good Data. These are no minus true for intelligence and defense applications than for commercial ones. The basic to arrest, procedure, and analyze information at scale, rapidly, is an important fact of the last decade [33].

There are approximately current applications of Map-Reduce such as Hadoop and Sphere most of which accept the original programming model and the structural design obtainable by Google. These structural designs attention on acting single stage Map-Reduce (calculations that included only one use of Map-Reduce) with better fault acceptance, and consequently stock most of the data outputs to some method of file system through the calculation. Additionally, in these runtimes, the repeated use of Map-Reduce produces new Map-Reduce jobs in each repetition filing or retrieving any static data continually. While these structures can be acceptable for single stage Map-Reduce calculations, they present substantial presentation expenses for numerous iterative uses [39].

5. Theory and methods about Hadoop

5.1 What is Hadoop and what it does?

At present, trade with datasets in the direction of terabytes or even petabytes is the truth. Hence, handling such large datasets in a well-organized method is a strong must for many users. In this situation, Hadoop Map-Reduce is a Big Data processing framework that has quickly developed a basic standard in both business and academic world. The key causes of such acceptance are the ease-of-use, scalability, and failover possessions of Hadoop Map-Reduce. On the other hand, these structures have a charge [40].

The performance of Hadoop Map-Reduce is commonly far from the fulfillment of a well-tuned parallel database. Thus, many research works (from business and academic world) have dedicated on developing the performance of Hadoop Map-Reduce tasks in several features. For instance, scientists have proposed altered data layouts, join algorithms, high-level query languages, failover algorithms, and query optimization techniques, and indexing techniques [40].

Hence, the previous year scientists have dynamically calculated the different performance difficulties of Hadoop Map-Reduce. Unluckily, users do not every time have a profound familiarity on how to efficiently abuse the different methods [40].

5.2 Why Hadoop

The requests for the storage system from the assignments can be précised as follows:

1. **Elasticity:** We want to be capable of improving additional size to our storage systems with a simple slide and no lost time. In some circumstances, we would need to increase capacity quickly, and the system must automatically stabilize the ability and use through new hardware [41].

2. **High write throughput:** Most of the applications store marvelous quantities of data and need considerable collective right amount [41].

   Efficient and low-latency strong consistency semantics within a datacenter: There are significant applications similar to Messages that need robust stability within a data center. This condition frequently rises straight from user potentials. For instance, 'unread' message amounts presented on the home page and the messages displayed in the inbox page view must be reliable with detail to each other. However, a worldwide spread powerfully reliable system is almost difficult [41].

3. **Efficient random reads from a disk:** In spite of the extensive use of application-level stores at Facebook scale, a lot of admissions oversight the store and hit the back-end storage system. MySQL is very efficient at arbitrary execution reads from disk, and any new system would have to be comparable [41].

4. **Performing High Availability and Disaster Recovery:** It is a service with actual great uptime to users that covers together calculated and unexpected actions. Additionally, we can accept the cost of a data center with nominal data loss and be able to serve data out of additional data center in an equitable time frame [41].

5. **Fault Isolation:** As we can see, running big farms of MySQL databases has displayed that fault separation is serious. Separate databases can and do go down, but any such occasion concerns only a minor part of users. Also, in a warehouse procedure of Hadoop, single disk failures influence only a small amount of the data, and the system rapidly improves from such errors [41].

6. **Atomic read-modify-write primitives:** Atomic additions and link-and-switch APIs have been precious in constructing open parallel applications and are needed from the primary storage system [41].

7. **Range Scans:** Some applications need an efficient recovery of a set of rows in a particular variety. For instance, all the latest 100 messages for a certain user or the hourly imprint totals over a day for a given promoter [41].

5.3 Hadoop Components

In addition to Hadoop itself, there are numerous open source projects built on top of Hadoop. Captain projects are defined underneath.

5.3.1 The Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) is a Java-based file system that affords scalable and reliable data storage that is planned to extend big clusters of service servers. HDFS, Map-Reduce, and YARN form the core of Apache™ Hadoop® [42].

5.3.1.1. What HDFS Does

HDFS was planned to be a scalable, fault-tolerant, distributed storage system that workings thoroughly with Map-Reduce. HDFS will "just work" beneath a variability of physical and systemic conditions. By allocating storage and calculation crosswise many servers, the mutual storage source can cultivate with the request while continuing efficient at every size [42].

These particular features confirm that the Hadoop clusters are very efficient and vastly offered:

- **Rack awareness**: Permits attention of a node's physical locality, when dealing with storage and scheduling jobs [42].
- **Minimal data motion**: Map-Reduce transfers calculate procedures to the data on HDFS and not the other way round. Processing jobs can follow on the physical node where the data exist into. These meaningfully decrease the network I/O designs and saves most of the I/O on the local disk or within the same rack and delivers very high total read/write bandwidth [42].
- **Utilities:** Analyze the condition of the file system and can rebalance the data on altered nodes [42].
- **Rollback:** Permits system operatives to get back the prior version of HDFS when an upgrade, in occasion of human or system errors [42].
- **Standby Name-Node:** Offers termination and supports excellent accessibility [42].
- **Highly operable:** Hadoop switches altered types of the cluster that might else need operative involvement. This project lets a single operator sustain a group of 1000s of nodes [42].

5.3.1.2. How HDFS Works

An HDFS cluster is included of a Name-Node which manages the cluster metadata and Data-Nodes that store the data. Files and directories are signified on the Name-Node by inodes. Inodes top score characteristics like consents, adjustment and admission times, or namespace and disk space shares [42].

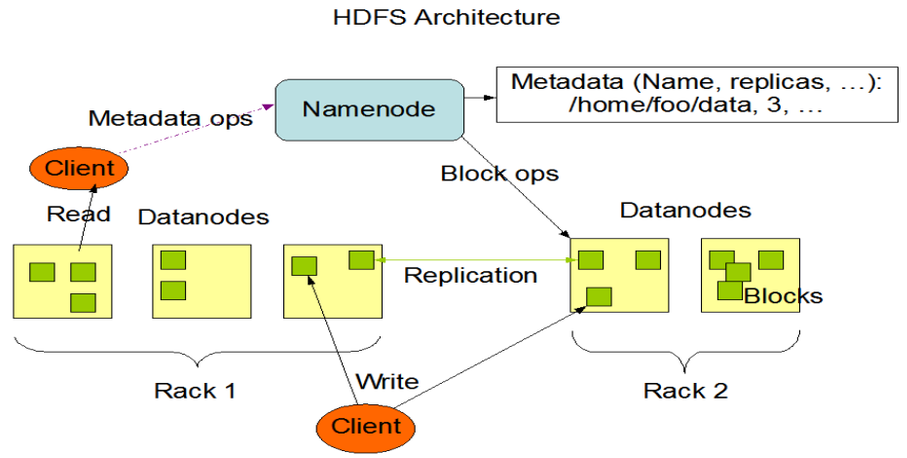

Figure 5.1: HDFS Architecture [42].

The file content is separating into big blocks (typically 128 megabytes), and for each block of the archive, it is individually virtual at several Data-Nodes. The blocks are kept on the local file system on the data-nodes. The Name-node dynamically observers the number of copies of a block. When a copy of a block is missing due to a Data-Node failure or disk failure, the Name-Node generates another version of the block. The Name-Node keeps the namespace tree and the mapping of blocks to Data-Nodes, holding the whole namespace image in RAM (Figure 5.1) [42].

The Name-Node does not straight direct requirements to Data-Nodes. It addresses information to the Data-Nodes by responding to pulses sent by those Data-Nodes. The commands contain commands to matching blocks to other nodes, eliminate local block copies, re-register and send an instant block report, or close the node [42].

5.3.2 NoSQL

NoSQL database management systems are different interactive database-management systems, in that they do not use SQL as their query language. The notion of these systems is that that they are superior to management data that doesn't fit simply into tables. They give out with the above of indexing, schema and ACID transactional possessions to generate big, virtual data stores for running analytics on low-cost hardware, which is valuable for dealing with shapeless data [43].

5.3.3 Hive

Hive is a data warehouse framework built on top of Hadoop, established in social media, cast-off for ad-hoc querying with an SQL type query language and as well as used for additional composite analysis. Users express tables and columns. Data is overloaded into

and saved over these tables. Hive QL, an SQL-like query language, is cast-off to make precise, reports, and analyses. Hive queries promotion Map-Reduce tasks. Hive is planned for set processing, not online business processing – unlike HBase, Hive does not deal real-time queries [44].

Therefore Apache Hive helps to evaluate Big Data by via the query language called HiveQL for the data source, like HDFS or HBase. The design is divided into Map-Reduce-oriented implementation, Metadata statistics for data storage, and a performance part that accepts a query from user or applications for implementation. To sustenance development by the user, it permits user state task at the scalar value, aggregation, and table level (Figure 5.2) [43].

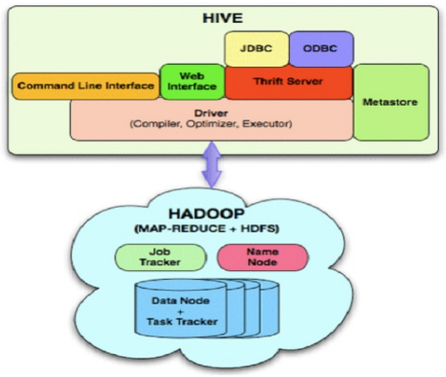

Figure 5.2: HIVE architecture [43].

5.3.4 Pig

Pig is a high-level data-flow language and affecting framework whose compiler produces structures of Map-Reduce programs for completing within Hadoop. Pig is planned for lot processing of data. Pig's infrastructure layer contains a compiler that fits Pig Latin programs into arrangements of Map-Reduce programs. Pig is a based client-side Java application, and users install it locally. Grunt is the Pig interactive shell [44].

5.3.5 Mahout and other expansions to Hadoop programming capabilities

Hadoop is not only for big-scale data processing. Mahout is an Apache project for constructing scalable machine learning libraries, with many algorithms developed on top of Hadoop. Existing algorithm effort parts of Mahout: clustering, classification, data mining and evolutionary programming. Apparently, the Mahout clustering and classifier algorithms have straight significance in bioinformatics - for instance, for clustering of big gene appearance data sets, and as classifiers for biomarker credentials [44].

Many bioinformaticians that procedure R might be attentive in the "R and Hadoop Combined Processing Environment." The Java package that mixes the R environment with Hadoop so that it is likely to code Map-Reduce algorithms in R. For the increasing public of Python users in bioinformatics, Pydoop, a Python Map-Reduce and HDFS API for Hadoop that permits full Map-Reduce applications to be written in Python, is offered. These are samples from the vast number of developers developing on further libraries for Hadoop [44].

5.3.6 Cascading

Cascading is a development implement a programming API for significant and affecting fault accepting data processing workflows on a Hadoop cluster. Cascading is a high, open source Java library that assembles on top of the Hadoop Map-Reduce layer. Cascading offers a query processing API that permits programmers to work at an advanced level than Map-Reduce, and further fast collecting complex spread procedures, and program them to build on dependencies [44].

5.3.7 HBase

Finally, a vital Apache Hadoop-based development is HBase, which is demonstrated on Google's Big-Table database. HBase enhances a dispersed, fault-accepting scalable database, manufactured on the top layer of the HDFS file system, with random real-time read/write access to data. Each HBase table is stored as a multidimensional spare map, with rows and columns, each cell having a time stamp. A cell assessment at a given row and column is exclusively recognized by (Table, Row, Column-Family: Column, Timestamp) Importance. HBase has its own Java client application interface, and tables in HBase can use together as an input source and as an output object for Map-Reduce tasks over Table-Input/Table-Output-Format. There is no HBase single point of failure. HBase uses Zookeeper, additional Hadoop subproject, for running of fractional failures [44].

Altogether table admissions are by the main key. Minor directories are likely done new index tables; programmers want to renormalize and duplicate. There is no SQL query language in base HBase. On the other hand, there is moreover a Hive/HBase addition job that lets Hive QL declarations contact to HBase tables for together analysis and introducing. Moreover, there is the free HBql development to improve a dialect of SQL and JDBC attachments for HBase [44].

A table is created from regions. Each region is definite by a start-Key, and End-Key may live on a changed node and is made up of some HDFS files and blocks, for each of which is simulated by Hadoop. Columns can be extra on-the-fly to tables, with simply the parental column relations being stable in a schema. For each cell is marked by column family and column label, so programs can continuously recognize what kind of data item a known cell has [44].

Furthermore to HBase, other scalable random access databases are now offered. Hadoop-DB is a mixture of Map-Reduce and a typical interactive db system. Hadoop-DB uses PostgreSQL for db layer Hadoop for communication layer, and the full version of Hive for a translation layer [44].

6. Hadoop Vendors and Frameworks

6.1 Vendors

According to design Big Data architecture, it is significant to understand the existing Big Data background and to combine it with the current frame. Furthermore, data management configurations are planned information or data served into the initiative combination tool, which moved the composed planned data into data warehouses or operating units. Then, altered analytical abilities were cast-off to expose the data. However, the different method of data management configurations that receive big data background is planned to encounter the velocity, volume, value, and variety of requests. Switch to these big datasets; different architectures have been designed that unite multi-node parallel processing systems [43]. Big data set has an additional sorting constructed on handling requests and qualified plans are offered for bunch processing and real-time processing [43].

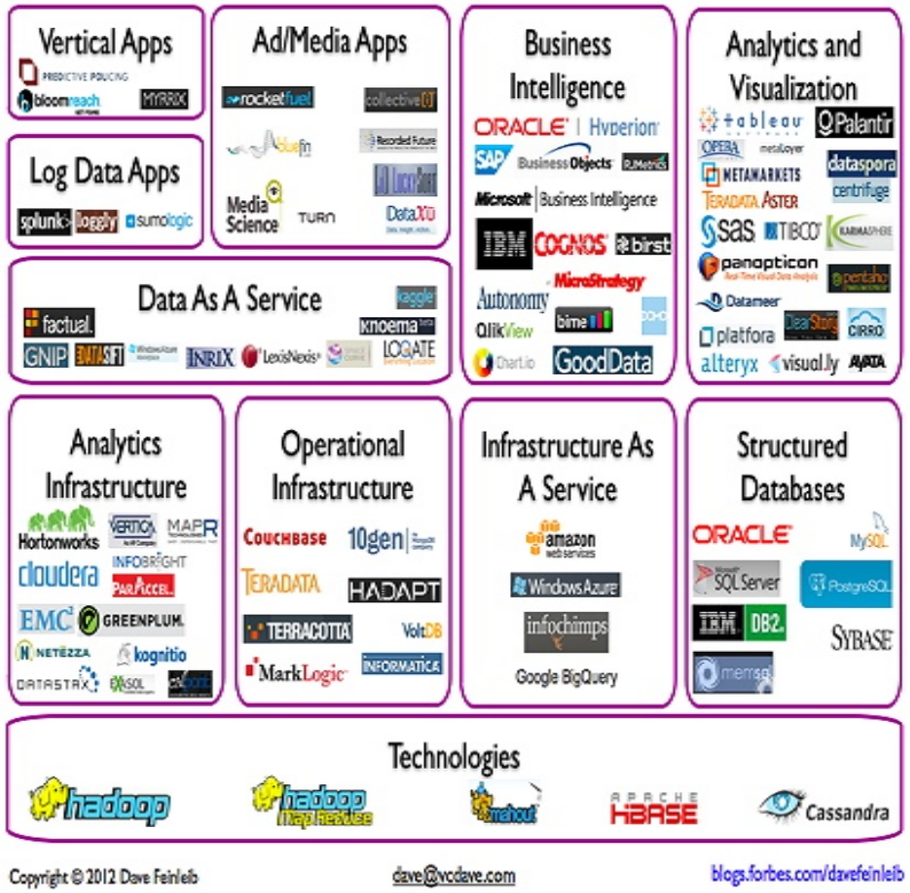

Figure 6.1: Big Data Landscape [43].

There is barely a vendor that doesn't have a big-data plan in sequence, with numerous corporations merging their branded database products with the open-source Hadoop tools as their approach to challenge velocity, variety, and volume. Several of the primary Big Data tools came out of open source, affectation a danger to customary IT vendors that have packed their software and reserved their intelligent possessions close to them. Though, the open-source environment of the development has also delivered a chance for traditional IT vendors, for the reason that creativity and management often find open-source tools off-putting [43].

As a result, traditional vendors have accepted Hadoop very warmly, packaging it into their trademarked systems so they can trade the effect to the enterprise as more contented and aware packaged results (Figure 6.1) [43]. Below we are seeing some major vendors which are based our experiments.

6.2 Hadoop Frameworks

6.2.1 Hortonworks

Hortonworks was developed by chief engineers from the Yahoo Hadoop software engineering team. In June 2012, the company launched a high-availability form of Apache Hadoop, the Hortonworks Data Platform on which it works together with VMware, as a result, was to aim corporations using Hadoop on VMware's vSphere. Teradata has also associated with Hortonworks to build products which aid their consumers to solve business issues in the new and improved way [43].

6.2.2 Cloudera

In this Book, we are implementing our Hadoop Cluster on Cloudera Framework. Cloudera is an initiative data management by offering the primarily united Platform for Big Data: The Enterprise Data Hub. Cloudera proposals Companies one place to store, process, and analyze all their data, authorizing them to spread the worth of current savings while allowing different methods to originate worth from their data [45].

Cloudera's platform, which is planned to precisely address consumer prospects and tasks in Big Data. It is offered in the usage of free or unsupported products (CDH or Cloudera Express, for those attentive only in a free Hadoop distribution). Alternatively, as protected, enterprise-class software such as Cloudera Enterprise - in Basic, Flex, and Data Hub edition, In the method of a yearly contribution [45].

The basic of Cloudera's platform, CDH, is open source (Apache License), so users continuously have the choice to transfer their data to a different. In fact, Cloudera is an open source vendor in Big Data, with its forces together underwriting more code to the Hadoop network [45]. Cloudera matches this open core with closed source controlling software that offers main creativity functionality demanded by clients such as maintenance for developing upgrades, assessing management, and disaster recovery. That software, still, does not store or process data and as a result, lock-in is not a problem [45].

However, we are using the Cloudera in this Book, because it is the basis for the next step of enterprise data architecture, one that compacts with the volume and difficulty of today's data. With Cloudera, we can store, process and analyze information in any format and at any scale. Involved in the necessary Apache Hadoop components, Cloudera is open source and contains all we need to create our 4-nodes cluster implementation for this Book. [46].

6.3 Reference to the Components: Cloudera Framework and RHadoop

6.3.1 *Cloudera Standard 4.8.5 Framework*

In our experiment, we are using the Cloudera Standard 4.8.5 Framework. It is very likely this version is no longer supported by the Cloudera vendor nowadays, but we used this specific version because it is a very stable, light and free Cloudera version, hence is ideal for our experiment.

6.3.2 Additional Components and Services of the Cloudera Framework

There is a group of extra software packages available to complement Hadoop, which deliver altered services built on the fundamental infrastructure, containing web access, SQL-like data querying, and management services [47].

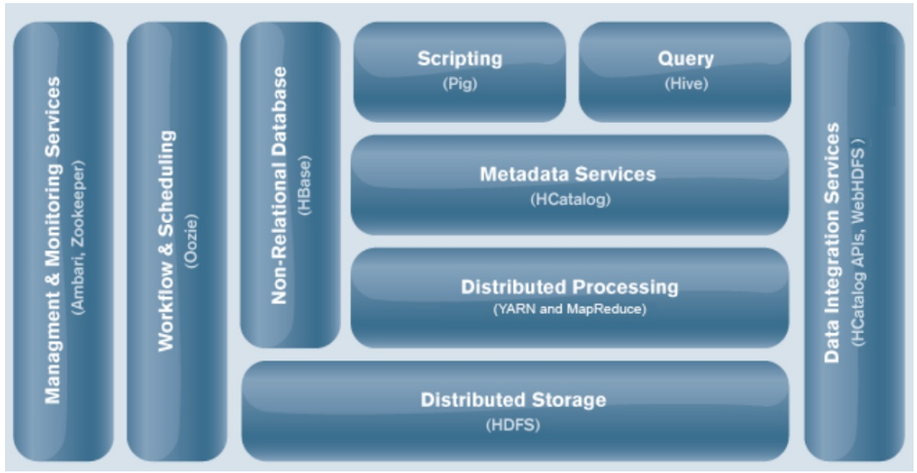

Figure 6.2: Cloudera Data Platform Architecture [47].

As shown in Figure 6.2 the most significant modules are the next and deliver main functionality:

- **Ambari** is a web-based tool for monitoring and managing Hadoop clusters. It supports HDFS, Map-Reduce, HBase, Hive/HCat, WebHCat, OOzie, Nagios, and Zookeeper [47].
- **HBase** is a distributed database that works on top of HDFS and offers significant table abilities [47].
- **Hive** is a data store on condition that data summarization and ad hoc querying. It offers a query language built on SQL, known as HiveQL [47].
- **HCatalog** is a storage controlling package for data. It grips the locality of the data and the metadata in a Hadoop cluster [47].
- **Pig** is a high-level scripting platform, which implements an interface to Map-Reduce in a practice scripting language called Pig [47].
- **Zookeeper** maintains an open-source spread structure and organization service for big distributed systems [47].

6.3.2 The R Language and RHadoop

The R language is an open-source programming language which focuses on statistics, data analysis, and data visualization. The R has plenty components, and It is highly extensible. Also, it is an object-oriented language and has strong capabilities on graphics. The R language is interpreted and is able to use it with a command line interpreter. Hence the R is available for Linux Operation Systems, also for Windows and Mac. In the end, a handy and famous development editor for R is the RStudio which we use it in our experiment [48].

The R language and the Hadoop framework both can combine very well. We can say there is an excellent match in Big Data analytics and visualization. Hence one of the most famous R packages to support Hadoop functionalities is the RHadoop, which has been developed by RevolutionAnalytics [48].

In our experiment, we are using the RHadoop components for the connection between the Cloudera - Hadoop framework and the R language. RHadoop is an excellent smooth and stable solution. Furthermore, RHadoop is a pack of five major packages of R, which can manage and analyze a Hadoop framework data [49]. Below at Table 6.1, we can see a short description of these packages.

Package Name	Description
rhdfs	This component gives a basic property to the Hadoop distributed classification system. The programmers of R will browse, read, write, and modify files, keep in HDFS from R inside. Installing this component separately on the node that may run the R shopper.
rhbase	This package provides basic property to the HBASE distributed info, exploitation the Thrift server. R programmers will browse, read, write, and modify tables keep in HBASE from inside R. Installing this package solely on the node that may run the R shopper.
plyrmr	This package permits the R user to perform common knowledge manipulation operations, as found in standard packages like plyr and reshape2, on big datasets to keep on Hadoop. Like rmr, it depends on Hadoop Map-Reduce to perform its tasks. However, it provides a well-recognized plyr-like interface whereas activity several of the Map-Reduce details. Installing this package solely each node within the cluster.
rmr2	A package that enables R developer to perform applied math analysis in R via Hadoop Map-Reduce practicality on a Hadoop cluster. Installing this package on each node within the cluster.
ravro	A component that provides the capability to read and write avro files from local and HDFS file system and gives an avro input format for rmr2. Installing this component only on the node that will run the R client.

Table 6.1: Description of RHadoop packages [50].

In the Annex Chapter, we can see step by step the setting up our Cluster, with 4 nodes based on the Cloudera framework also the setup up of the RHaddop component as well as the environment of the R language.

7. The Skyline Algorithms and Map-Reduce

7.1 The Skyline Algorithms

Recently, there is a growing interest in Skyline queries. The Skyline of a set of points is defined as the points that are not dominated by any other aspect. A point x dominates another y point if x is no worse in any dimension (than those of interest) by y and is better at least one. For example, if someone is interested in a cheap hotel near the beach the Skylines will contain all the answers. The solution will not include the cheapest hotel if it is too far or the closest to the beach if it is too expensive. Satisfactory algorithms have already been proposed for implementing Skyline queries on a traditional basis. Unfortunately, however, most of them are based on assumptions that do not apply to P2P systems [49].

Researchers propose to extend the SQL language so that it can support Skyline queries, present and measure the efficiency of alternative algorithms for implementing Skylines, and show how this function can be combined with other functions on the base (e.g., join and Top N). Based on these views as well as on other researchers we will implement a skyline algorithm in querying multidimensional [49].

7.2 Skyline Algorithms based on Map-Reduce

According to K.Mullesgaard *et al.,* previous tasks have provided several ways to execute skyline queries using Map-Reduce without using the parallel processing advantages since a significant part of the query is running sequentially, and only one reducer is used to calculate the final skyline points [51].

So, they propose a new algorithm, the Map-Reduce - Grid partitioning Multiple Reducers Skyline (MR - GPMRS), which runs the entire query in parallel. The basis of this algorithm is a breakdown of the initial set of data using a technical grid. A bitstring is used; recording which partitions is not a quote, which makes it easy to make decisions based on the whole of the total data. The way in which the function of the territory roles in the skyline, as well as in skyline querying with grid-based spatial resolution, allows the data to be shared in parts that can be processed independently of one another. This means that different reducers can handle these data parts. Those have as a result that MR-GPMRS works very well for large sets of data and big clusters. In their experiments, MR-GPMRS runs several times better than real-time algorithms to a significant set of data. However, the fact that the reducers need to take copies of the apartments increases the cost of communication significantly. Their experimental study shows that the use of multiple reducers is not the best choice when the skyline is low. Thus, to optimize the algorithm for any set of data, it is necessary to find a way, that the MR-GPMRS algorithm to choose the number of reducers intelligently it will use [51].

L.Chen *et al.* they present a new MapReduce Skyline method for parallel processing of skyline queries. This new method significantly reduces the time of the Reduce phase due to the limitation of the unnecessary costs for the existence of dominated points. The experiments carried out at Hadoop showed that their method works well with the increase in dimensions and the plurality of data. Also, they revealed that the new angular partitioned Map-Reduce method runs 1.7 and 2.3 times faster than dimensional and grid partitioning methods [52].

B.Zhang *et al*. define the problem of processing skyline queries in the MapReduce framework. Based on different data partitioning strategies, they develop three algorithms for MapReduce skyline mapping: MapReduce based BNL (MR-BNL), MapReduce based SFS (MR-SFS), and MapReduce based Bitmap (MR-Bitmap). Extensive experiments have been carried out to evaluate and compare these alloys under different conditions, such as data distribution, dimensions, buffer size and cluster size. The experiments showed that the MR-BNL and MR-SFS algorithms function are well in most cases when they encounter problems with dimensions in parallel environments. The MR-Bitmap algorithm functions the same regardless of the data sets, especially when the bitmap matches exactly the memory of a single node [53].

In this book, the skyline algorithm is based on our own implementation Map-Reduce method and is simpler than the above algorithms. Because of the simplicity of our algorithm we naming it, Map-Reduce Simple Skyline Algorithm (MR-SSA). Below we are studying the algorithm function more extensively [49].

7.3 Structure and Implementation of the MR-SSA

7.3.1 Design the algorithm

The next step after designing the algorithm is its development. The development of the algorithm is implemented on the R studio with the use of R language. Also, it is fundamental the installation of the libraries of the packages rmr2, rJava, and rhdfs and before running the algorithm and to define the correct paths were called the libraries of Hadoop Framework. All those steps are necessary for the proper functioning of the algorithm [49].

The same algorithm also includes the production the random number generator code. In that code, the user defines the number of points to be created, the maximum value they can get, and the file name. After the code is executed, the created files are placed in the local file system [49].

7.3.2 Parameter Settings and Hadoop File System

When we are writing the code, a goal that had to be achieved was its versatility when it was run in different scenarios, without the user having to process the code every time. In this case, the initialized variables are the highest value that a coordinate can take the number of partitions and the number of dimensions of the points. Hence, whenever a user wants to change a parameter, it is enough to process only that particular file and not the entire code [49].

This particular file must be located in a local folder of the virtual machine so that the code can read it. According to exploit this file, a function has been created, which reads the file. In addition to setting the algorithm parameters, the HDFS must also be configured. Initially, the user must create the HDFS path to upload the original files for editing [49].

7.3.3 The Logic Diagram of the MR-SSA

In Figure 7.1 we see the logic diagram of the algorithm we have developed. As we observe, it is in full development and does not include only the three core parts of the algorithm we have mentioned in previous chapters, but the entire flow of our algorithm. As we observe, our algorithm begins with the function of creating an arbitrary dataset of 2000 points that we have defined, and then we select the dataset with which we want to start our process. In

parallel and independent of the previous two processes we define the paths for Hadoop Libraries and load the appropriate libraries that work with R to be able to run Map-Reduce processes [49].

Then it transfers our dataset to DFS and then starts the time measurement function where it is useful for measuring the processing performance and comparing each dataset to each node cluster.

After that, we start the first Map-Reduce process (job), where it is the ordering function of our dataset in ascending order along the points in conjunction with the columns. This process is important because one must precede the final finding of the points from the next process of finding the skyline points and on the other hand that if this process is not carried out, the skyline points which will found will not be the right [49].

After that, the second job begins, which as input takes the results of the first job. At this stage is the finding of the skyline points. This phase is once again a Map-Reduce job process [49].

Once the skyline points have been found, we are finishing the timing function. Then we show the skyline points where they are our results, and finally, we optionally display them in corresponding graphs so that we also have a graphical representation of our results on the axes we set in the beginning [49].

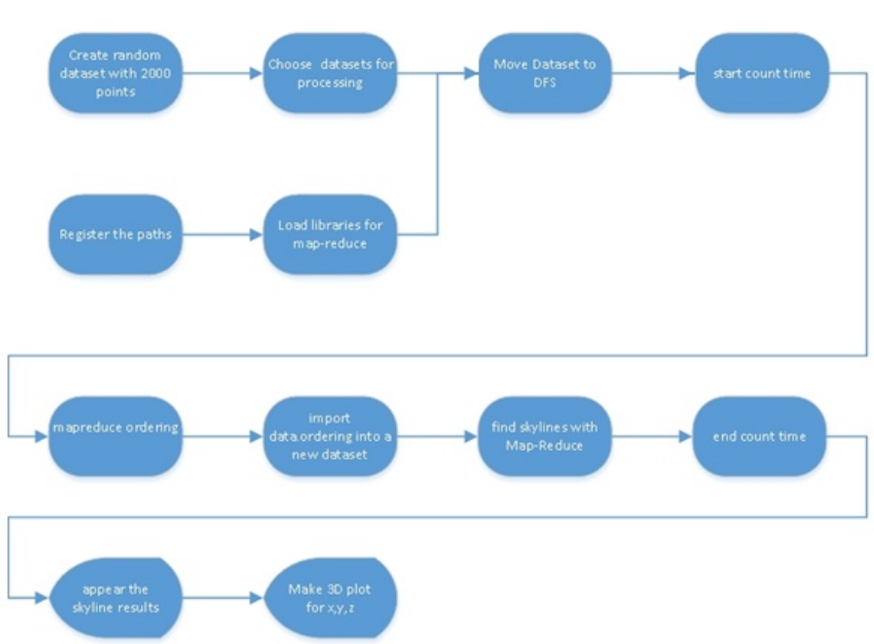

Figure 7.1: The logic diagram of the MR-SSA [49].

So, as we can see below, the basic functional structure of the algorithm we developed in the form of pseudocode. Hence, we focus on the three key parts of the algorithm such as:

1. **Creation of random numbers up to 2000 points.**

   The random point creation is vital because it feeds with raw data our algorithm for processing. Each time we run the algorithm, the data will be different and continually updated.

2. **Sort the dataset with the MapReduce process.**

   The process followed by Map-Reduce classifies the dataset with optimal algorithms, to quickly index skyline points [49].

3. **Finding skyline points with the MapReduce process**

   The process which follows from MapReduce helps to locate the skyline points effectively and to present the result optimally [49].

7.3.3.1 Pseudocode Structure of the MR-SSA

Input: The generated dataset **Dataset.points**
Output: The skyline dataset **nondominated**

Create random dataset with 2000 points for 6 dimensions (columns)

x1 = Make sample 2000 points
y1 = Make sample 2000 points
z1 = Make sample 2000 points
x2 = Make sample 2000 points
y2 = Make sample 2000 points
z2 = Make sample 2000 points

Map-reduce ordering

Map task

For each point k in **Dataset.points**
Compute v subspace flag keyval
Output(v,1)

Reduce task

For each subspace flag keyval
Compute **order** (v)
Output (**Skyline.points)**

Map-Reduce skylines

Map task

For each point k in dataset **Skyline.points**
Compute v subspace flag keyval
{
Compute skylines points in y
}
Output (**nondominated**)

Appear the skyline results

Nondominated

Make 2D plot for x1,y1,z1,x2,y2,z2

Plot **nondominated** [49]

8. Method and Execution of the Experiment

8.1. Experimental Part Specifications

8.1.1. Big Data Experiment with 4 nodes cluster and 5 columns datasets

In this chapter, we are focusing on our experimental part of this book. The experimental part of our study begins with the introduction of the entire algorithm on the R Studio command line and completes in series per function with the ultimate result of finding skyline points and finally, displaying graphs.

The experimental part takes place in a virtual 4-node computing cluster. The whole Hardware infrastructure we are using is a local PC with the total shared memory of 14GB, 6 Logic CPUs and a Solid State Hard Disk. Finally, we are using four virtual machines nodes, with Linux Operating System (Table 8.1).

The Software we are using for our experiment including the R language which is based our Map-Reduce Simple Skyline Algorithm (MR-SSA). Also, we are using the Cloudera Standard 4.8.5, a Big Data platform which is running on the Linux Centos 6.6 Operating System. Hence the host Operating System is the Windows 10, finally, the virtual machine application is the VirtualBox 5.1.8 (Table 8.1).

Hardware	4 Nodes Cluster Specifications
Total shared memory: 14GB	1st Node: 7GB Ram - 2CPUs
Logic CPUs: 6	2nd Node: 2.3GB Ram - 2CPUs
SSD Hard Disk	3rd Node: 2.3GB Ram - 2CPUs
4 VM Linux Linux nodes	4th Node: 2.3GB Ram - 2CPUs
Software	**Datasets**
R language	Dataset with 2000 random points
Map-Reduce Simple Skyline Algorithm (MR-SSA)	Dataset.1: 2 columns dataset
Cloudera Standard 4.8.5	Dataset.2: 4 columns dataset
VM OS: Centos 6.6	Dataset.3: 6 columns dataset
VM Application: VirtualBox 5.1.8	Dataset.4: 8 columns dataset
Host OS: Windows 10	Dataset.5: 10 columns dataset

Table 8.1: Hardware, Software, and Datasets specifications.

Our experiments are taking place on a 4-Nodes computing cluster. Below in Table 8.2 as we can see, we have included the running times in seconds of the five Datasets, with two, four, six, eight and ten columns of data, respectively. Furthermore, we have for each Dataset five-time measurements. Finally, in the last row of our table, we are including the average of the running time measurements which is a benchmark for conclusions for our experiment.

For each Dataset, we deliver five metrics. We do this because we want to show as much as possible what sample of measurements for each Dataset. Our metrics are actually a time in seconds that we need to execute on the Map-Reduce Simple Skyline Algorithm (MR-SSA). In this case, we are starting the time after transferring the dataset to the Hadoop file system and before starting the first Map-Reduce process which is the order of the points.

Once we complete our measurements for each Dataset, we take a score at the end of the average time of our five metrics in seconds (Table 8.2). Finally, after we calculate all of their mean limits and we compare them for conclusions (Figure 8.1 & Figure 8.2).

It is good to mention that we observe during in our experiments if we conducted our experiments with datasets over 2000 points (i.e., 5000 points), we noticed that the available main memory was depleted and the system pumping the swap memory from the Hard Disk space. That has a result, a very slow process calculations, resulting in their rejection after a long time and of course the collapse of the Map-Reduce process. This is due to the memory limitation; the datasets we are using are 2000 points for each dimension. In this case, to simulate a large volume of data, we exponentially increase the datasets dimensions, in many columns datasets.

Datasets	1st Measurement	2nd Measurement	3rd Measurement	4th Measurement	5th Measurement	Average Time
2 Columns Dataset	108.628 sec	104.448 sec	109.189 sec	103.176 sec	103.950 sec	105.8782 sec
4 Columns Dataset	238.385 sec	229.264 sec	248.095 sec	257.275 sec	270.502 sec	248.7042 sec
6 Columns Dataset	348.733 sec	362.063 sec	359.867 sec	336.138 sec	309.883 sec	343.3368 sec
8 Columns Dataset	502.285 sec	500.048 sec	482.066 sec	473.650 sec	491.366 sec	489.883 sec
10 Columns Dataset	591.871 sec	587.234 sec	605.330 sec	587.944 sec	606.555 sec	595.7868 sec

Table 8.2: Measurement results with 4 nodes for 5 datasets.

8.1.2. Time ranges and comparisons

Based on our measurements for each dataset, we receive the following graphs. As we can see in Figure 8.1, we observe the times of each measurement from each dataset.

In the below graph (Figure 8.1) as well as in chart (Figure 8.2), for the two columns dataset, we see that our metrics are very close to the 100-second range. In the four columns dataset, we see that the measurements are about 250 seconds. In the third dataset (six column dataset) we see that the times are round in the 350 second area. Finally, for the eight and ten columns datasets, the times are about 500 and 600 seconds respectively.

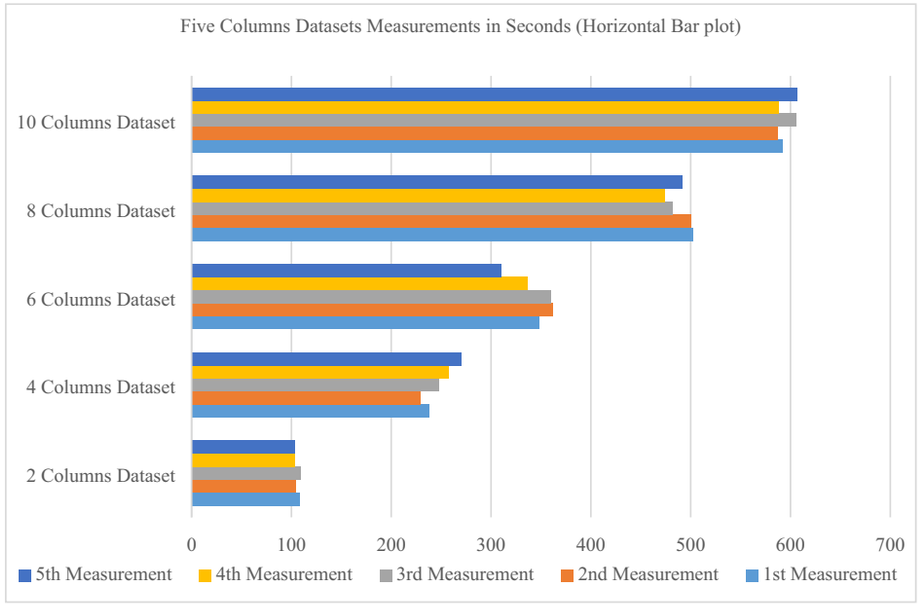

Figure 8.1: Five Columns Datasets Measurements in Seconds (Horizontal Bar plot)

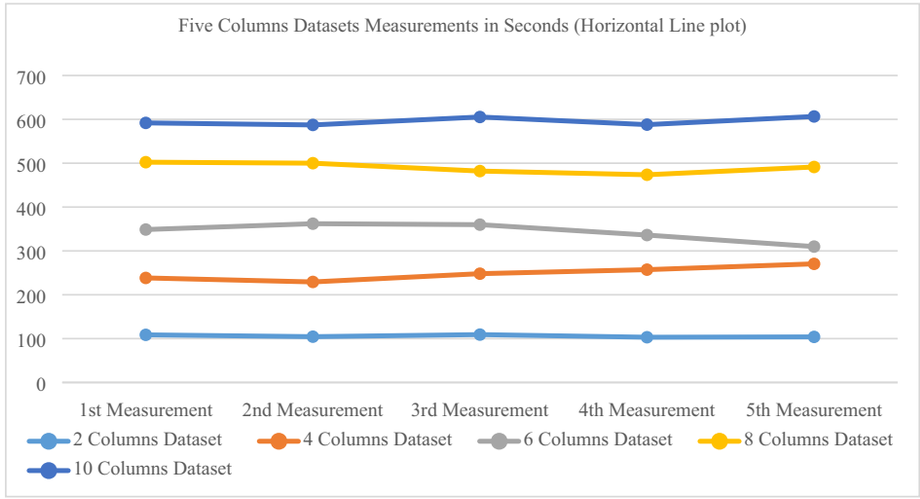

Figure 8.2: Five Columns Datasets Measurements in Seconds (Horizontal Line plot).

In the following graphs (Figure 8.3 and Figure 8.4) we observe the average time of our measurements in seconds. As we can see from the graphs as we increase the columns by two, we observe an exponential growth in each dataset's time. Also, in Figure 8.4 we observe the growth in the time between the datasets, and as we can see that it follows an exponential curve.

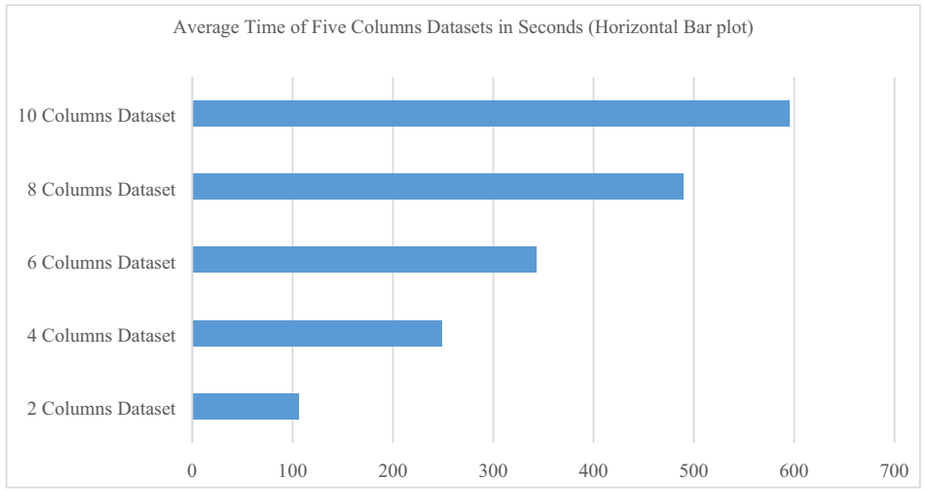

Figure 8.3: Average Time of Five Columns Datasets in Seconds (Horizontal Bar plot).

Thus, we notice that the 2nd dataset is 2.5 times the length of time relative to the 1st Dataset. The 3rd Dataset is about 1.4 times slower than the 2nd Dataset, and the 4th Dataset is about 1.4 times slower than the 3rd Dataset. Finally, the 5th Dataset is about 1.2 times slower than the 4th Dataset, almost reaching the 600 seconds.

As we can see from the above, we can say that as the dimensions of a dataset exponentially increasing, the difference in time between the two last dimensions decreasing to a degree of measurement. These may mean that a Hadoop-based computing cluster is better to process as much as we get a larger volume of data as the differences in data processing times from one point to the next are the same.

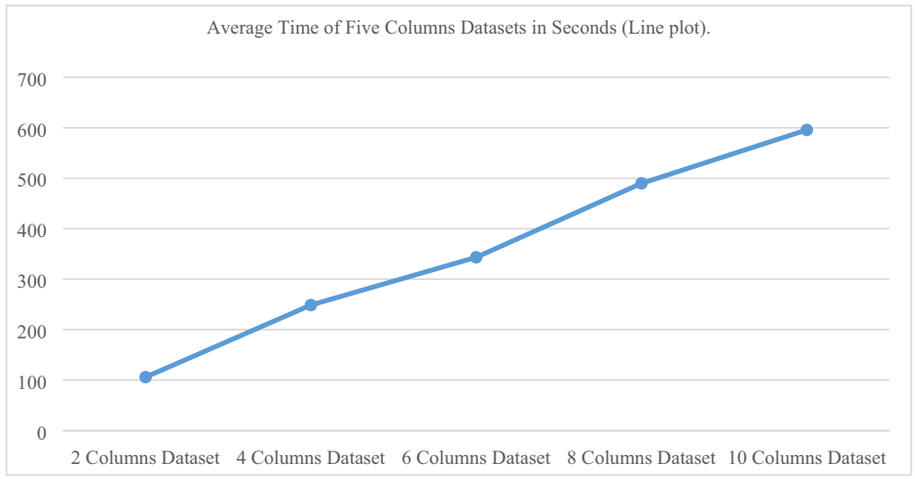

Figure 8.4: Average Time of Five Columns Datasets in Seconds (Line plot).

So according to the above, we draw the following conclusions:

- Increasing time over the previous measurement is all-and-too-small to the extent that we increase the volume of data.
- The average of the datasets time increases exponentially.
- In small datasets, the reduction in times as nodes grow is small to negligible.
- To big datasets, the decline in time as we increase the data yields is starting to be significant.
- As bigger is the dataset, the higher the profit in time.

9. Conclusion

9.1 The Goals of this Book

The goals of this book are to study and analyze the fundamental problems and needs of processing a big amount of data. Also to examine at a basic level, the methodology of Map-Reduce algorithms and to process them in the Hadoop framework. After that to study the methods and the benefits of the Hadoop framework. Hence, we make a reference to the Hadoop components.

Furthermore, we focus on the Hadoop vendors and frameworks. From those Vendors, we choose for our experiment the Cloudera platform for our Map-Reduce experiments and also we study the RHadoop components where these components are used for the connection between Hadoop framework and R language.

9.2 The Need for Implementation of Big Data

As we have seen from Chapter 1, the need to process large volumes of data is necessary for more and better accuracy of results and conclusions from Big Data. The category of Big Data includes and including and from another kind of data to a greater or lesser extent. The uses of Big Data as well as their purposes are sufficient and are applied to a plethora of implementations.

9.3 Map-Reduce, Algorithms and Other Methods

As we have seen from the previous chapters, there are many cluster systems which can process parallel data. These cluster systems can run the processes only locally, and in small-scale clusters, these systems are the OSCAR, Rocks, and OpenMosix.

The Map-Reduce approach is big scale (and can be even greater) and does not require parallel based applications to run. Also, can be used for every kind of data. We can say Map-Reduce is more efficient in parallel processing and the data are storing in the HDFS. Hence can execute the code in Java and also in other languages such as C, C++, C# and scripting languages like PHP, Python, and Perl. In this book as we see, we develop the MR-SSA which is based on R language.

Furthermore, the programs are automatically become parallelized and running on a big cluster of Hardware. Hence, the Map-Reduce approach is used from big industry vendors like Facebook and Yahoo. In the end, the Map-Reduce approach is used for our Map-Reduce Simple Skyline Algorithm because as we mentioned Map-Reduce is a highly scalable method and can be utilized through several computer nodes.

9.4 Hadoop and Implementation

The Hadoop framework is essentially the Map-Reduce method. The reason for this is to facilitate the use of Map-Reduce in applications. The Hadoop framework also consists of several components such as the Hadoop Distributed File System (HDFS), which is very important for storing and processing parallel data. Additional useful components are the NoSQL Database, Hive, Pig, Mahout, Cascading, and HBase.

Finally, a report we can do is in the Hadoop frameworks where the best known are Hordonworks and Cloudera. In our experimental part, we are using the Cloudera framework

in which we have installed the RHadoop component together with the components rhdfs, rhbase, plyrmr, rmr2, and ravro, for connecting the R language to the Cloudera framework.

9.5 The Map-Reduce Simple Skyline Algorithm (MR-SSA).

As we have seen in Chapter 7, there are several Skyline algorithms based on the Map-Reduce method. Some of these algorithms are MR-GPMRS, MR-BNL, MR-SFS, and MR-Bitmap. The MR-SSA approaches the philosophy of the BNL algorithm but has a simpler structure and it is based in our study.

By making a brief description, the basic implementation of our algorithm begins with the creation of datasets with 2000 random points, then the Dataset processing option is selected, and in the next phase, the algorithm transfers the dataset to the Hadoop File System (HDFS).

The main part of our algorithm begins with the sorting of random points and then processing the Skyline points. Finally, we show the results of the Skyline points and optionally display some relevant graphic for them.

9.6 Prospects of Using of the MR-SSA

The chances of using the Map-Reduce Simple Skyline Algorithm of this book are many, some of which could be for:

- The development of the algorithm to find nearby ports on a coastline [49].
- The quick and instant finding of airfares by comparing prices and services [49].
- To find planets in our galaxy with some specific features we have set [49].

We can also point out the application of our algorithm for academic use such as:

- Use our code to use as a reference for multi-node and hardware time measurement [49].
- Comparing methodology with different technologies and parallel programming approaches [49].
- Compare the odds among the various versions of the software that was based [49].
- Comparing different approaches and skyline algorithm yields with Map-Reduce implementation [49].

Finally, our algorithm could be applied in areas like:

- In Economics: Several issues involve multiple objectives alongside constraints on what combos of these objectives area unit come-at-able [54].
- In Finance: A standard drawback is a decision on a portfolio once their area unit two conflicting objectives. The need to own the arithmetic mean of portfolio returns be

as high as attainable, and also the need to own risk. Typically is measured by the quality deviation of portfolio returns, be as low as attainable [54].

- In Optimal control: In engineering and social science, several issues involve multiple objectives that are not expressible as the-more-the-better or the-less-the-better; instead, there's a perfect target worth for every objective, and also the need is to urge as shut as attainable to the specified worth of every objective [54].

- In Optimal design: Product and process design can be largely improved using modern modeling, simulation and optimization techniques [54].

- In Electric power systems: Reconfiguration, by exchanging the practical links between the conditions of the system, represents one in all the first necessary measures which might improve the operational performance of a distribution system [54].

9.7 Future Expansions

As we have seen in the previous chapters, the experimental portion of the work was mainly based on VMs in which the overall infrastructure and algorithm were implemented. While as we have seen, we did our experimental part with four computer clusters. We also saw that the experimental part ran with some specific hardware on a local computer using virtual machines. Additionally, concerning the data we ran, we had a 2000-point restriction [49].

According to the above, future developments of the MR-SSR algorithm could be several, such as:

- Setting up and running on Cloud Computing [49].

- Increasing the nodes [49].

- Running multiple parallel processes [49].

- Running the algorithm for more than 2000 points [49].

- Running the algorithm and timing it into new technologies such as Apache Spark [49].

- Set up and execute the algorithm in different programming languages (e.g., Java or Python) [49].

- Further refinement of the MR-SSR algorithm itself by other techniques, such as writing and collecting results in a local file and not so extensively in the main memory of the cluster computing [49].

10. Annex

In this Chapter, we see an analytical tutorial about the setup of the 4-nodes cluster on Virtual Machines with the VirtualBox application. We continue with the installation of the Cloudera framework and the parameterization of the Master Cluster and Slave Clusters. Furthermore, we see the setup of the RHadoop components in Cloudera Framework. Finally, we see the complete Map-Reduce Simple Skyline Algorithm in R language, and we are executing it in the R Console.

10.1 VirtualBox installation

1. We begin the installation with the latest version of the Oracle VM VirtualBox.

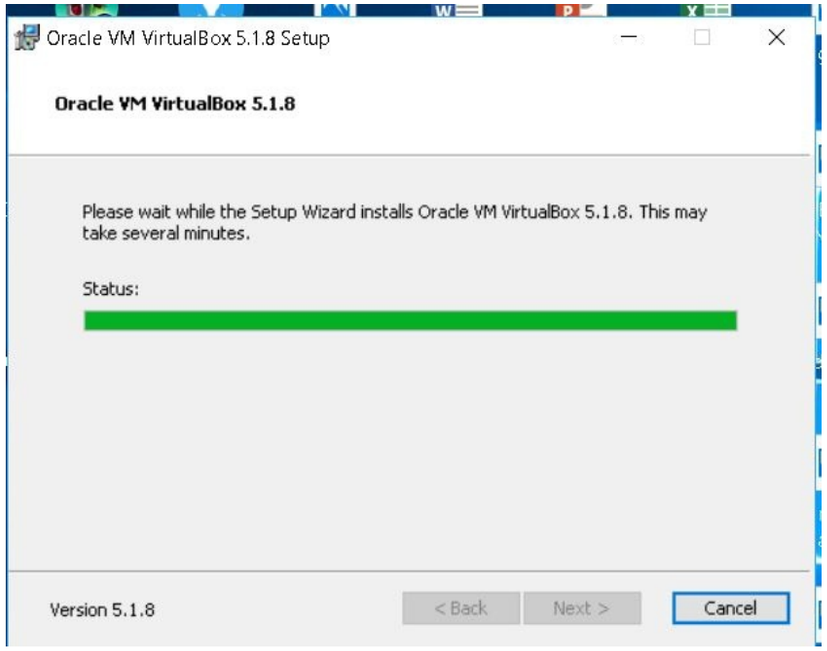

Figure 10.1: VirtualBox Installation-part1.

Figure 10.2: VirtualBox Installation-part2.

2. After the installation of the VirtualBox, we set up the 4-nodes cluster.

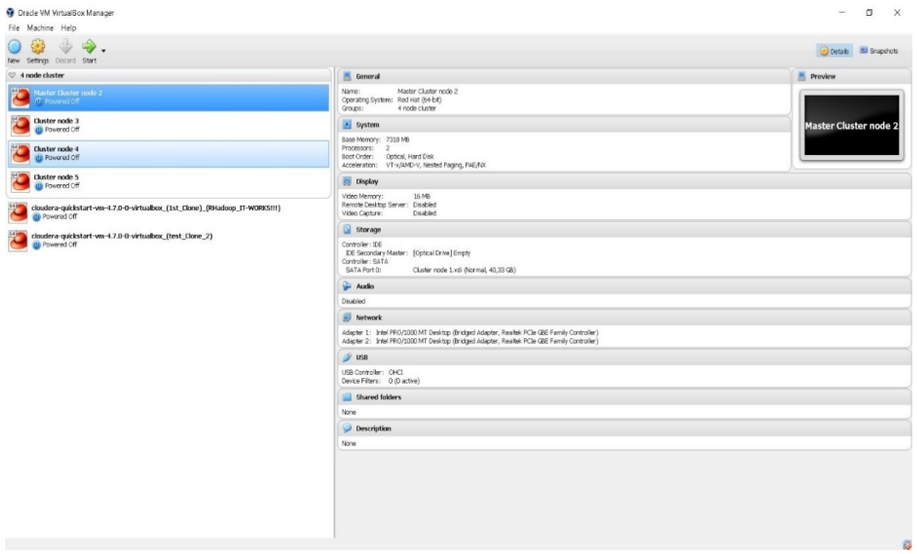

Figure 10.3: 4-Node Cluster Setup in VirtualBox

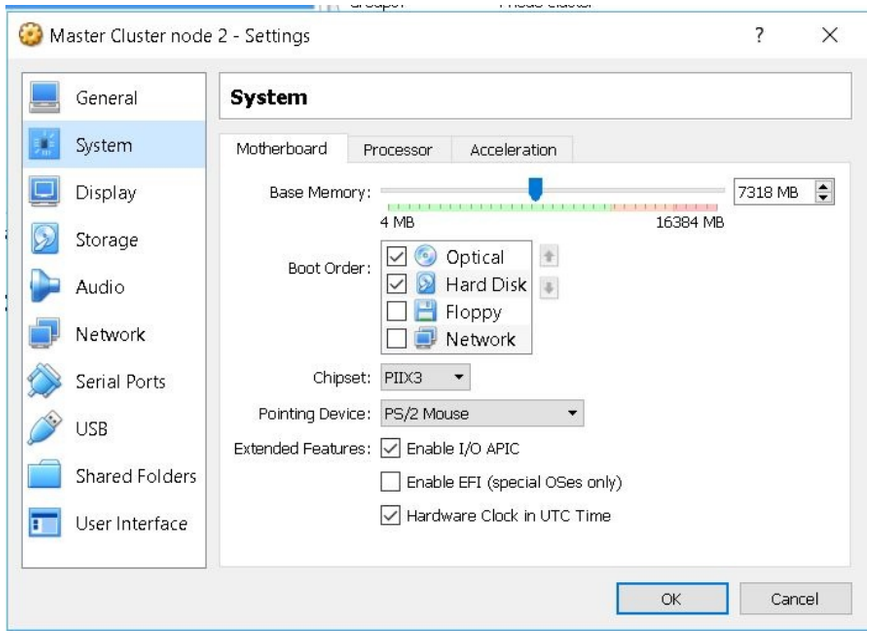

Figure 10.4: Master Cluster Node Setup in VirtualBox Configuration System-part 1.

3. We configure the Memory for the Master Node. 7GB RAM or above is fine for a small scale cluster running the core Hadoop components in Cloudera Framework.

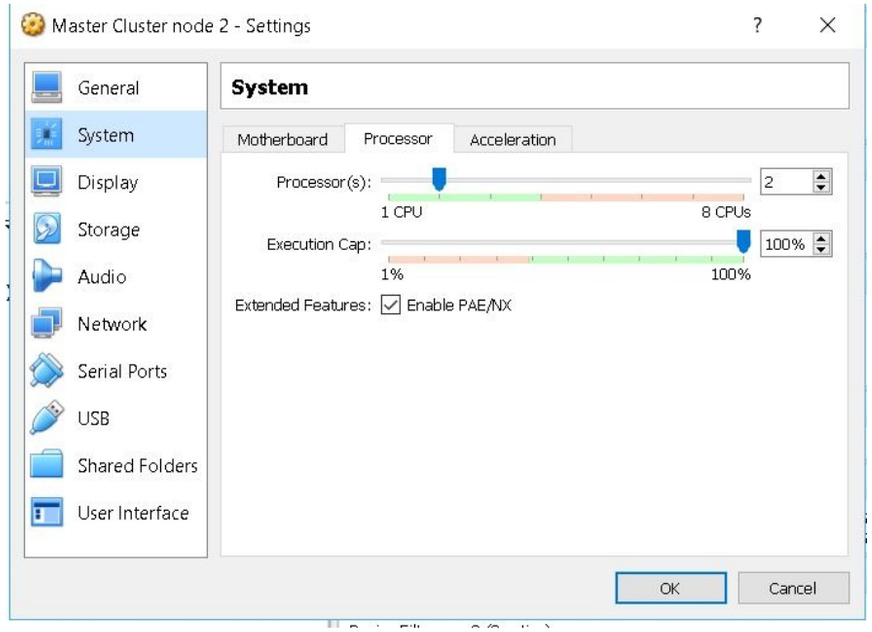

Figure 10.5: Master Cluster Node Setup in VirtualBox Configuration System-part 2.

4. We configure the CPU for the Master Node. In our experiment, we use 2 VM Processors.

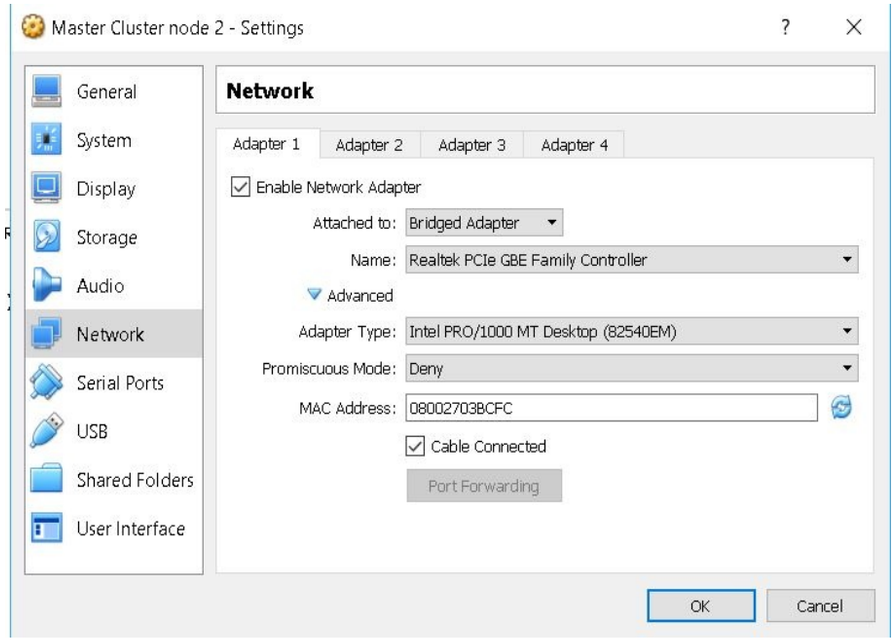

Figure 10.6: All Cluster Node Setup in VirtualBox Configuration Network.

5. We are making the configuration network for all Nodes. We select the *Bridged Adapter* as we see in the Figure 10.6.

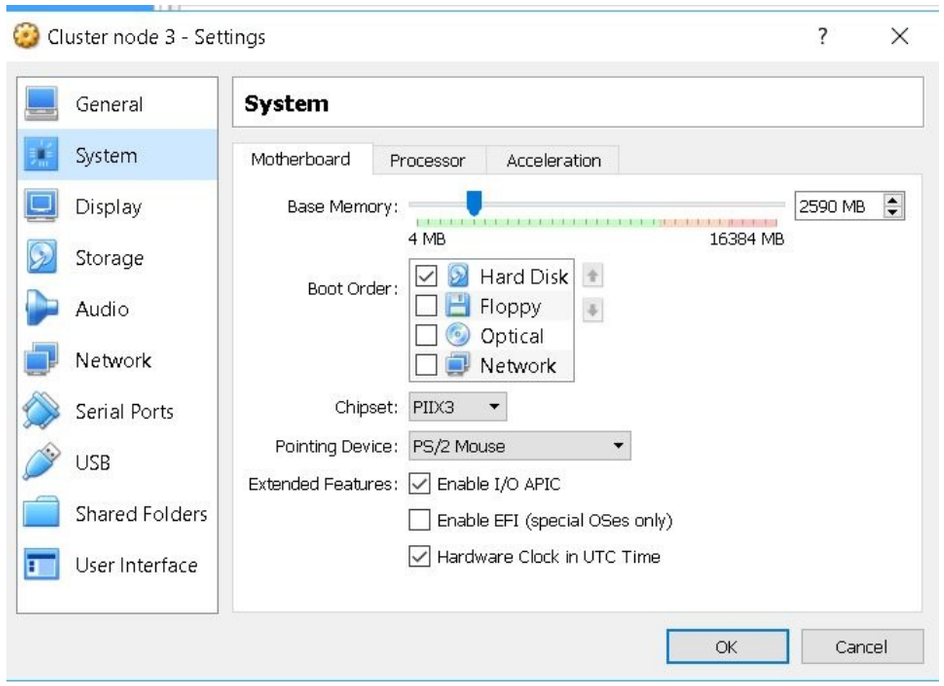

Figure 10.7: Slave Cluster Node Setup in VirtualBox Configuration System-part 1.

6. Making configuration for the slaves' Node Memory. 2.5GB RAM or above for each slave node is fine for the small-scale cluster.

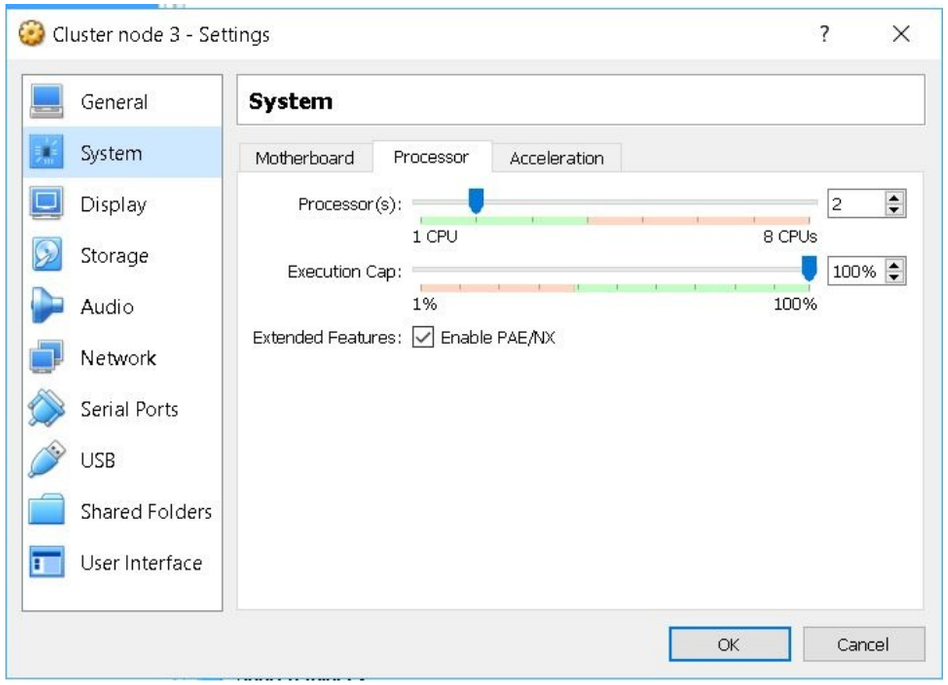

Figure 10.8: Slave Cluster Node Setup in VirtualBox Configuration System-part 2.

7. Making configuration for the slaves' Node CPU. 2 VM Processors for each slave node is fine for one-time processing.

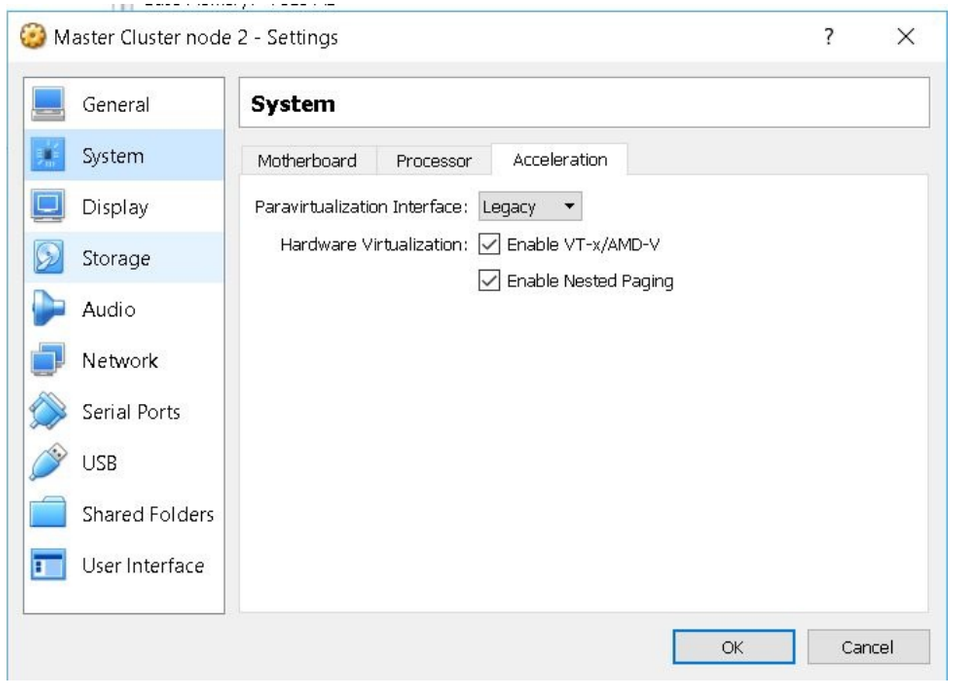

Figure 10.9: Cluster Node Setup in VirtualBox Configuration System-part 3.

8. It is necessary to check the option *Enable VT-x/AMD-V* and the CPU to support the Virtualization Technology.

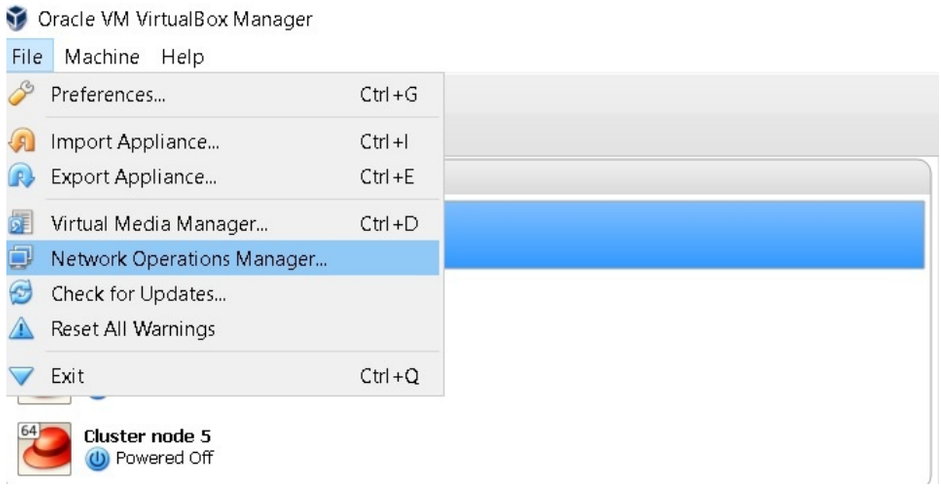

Figure 10.10: 4-Node Network Setup in VirtualBox Configuration-part 1.

9. Finally, as we can see from the Figure 10.10 to Figure 10.12, we make the overall network configuration for the 4-nodes Cluster.

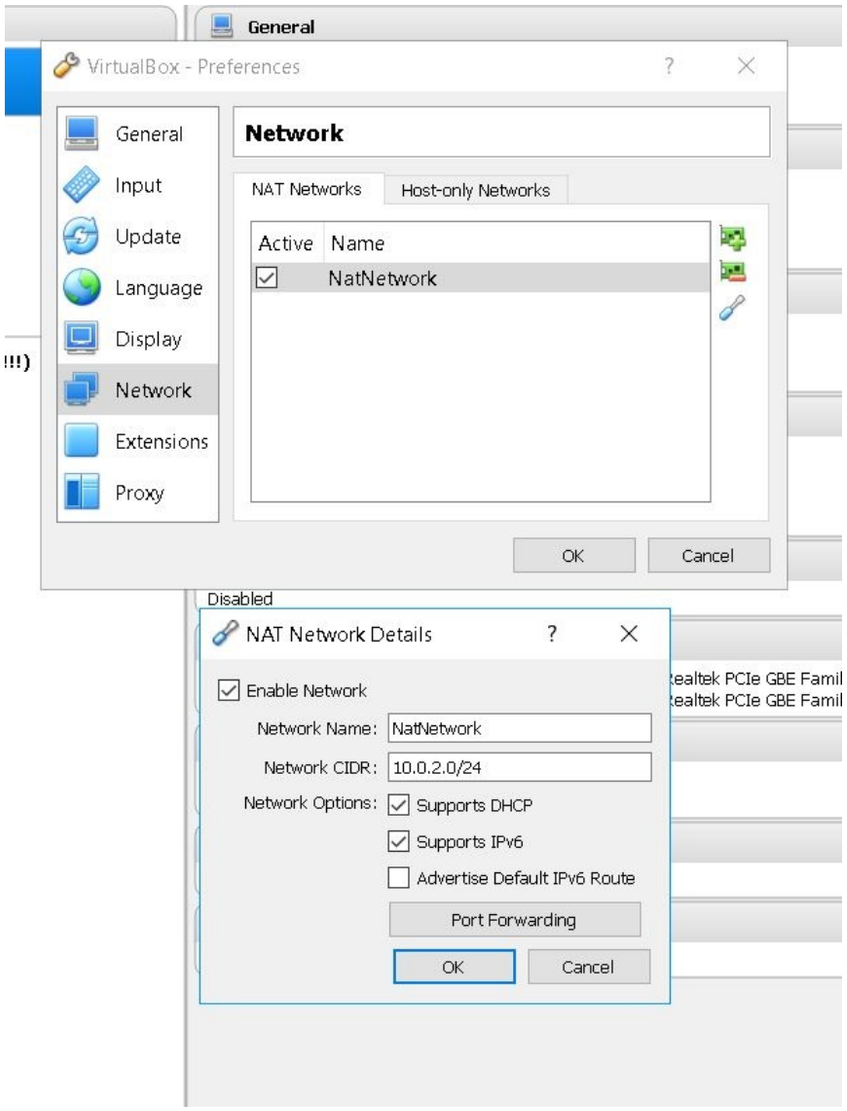

Figure 10.11: 4-Node Network Setup in VirtualBox Configuration-part 2.

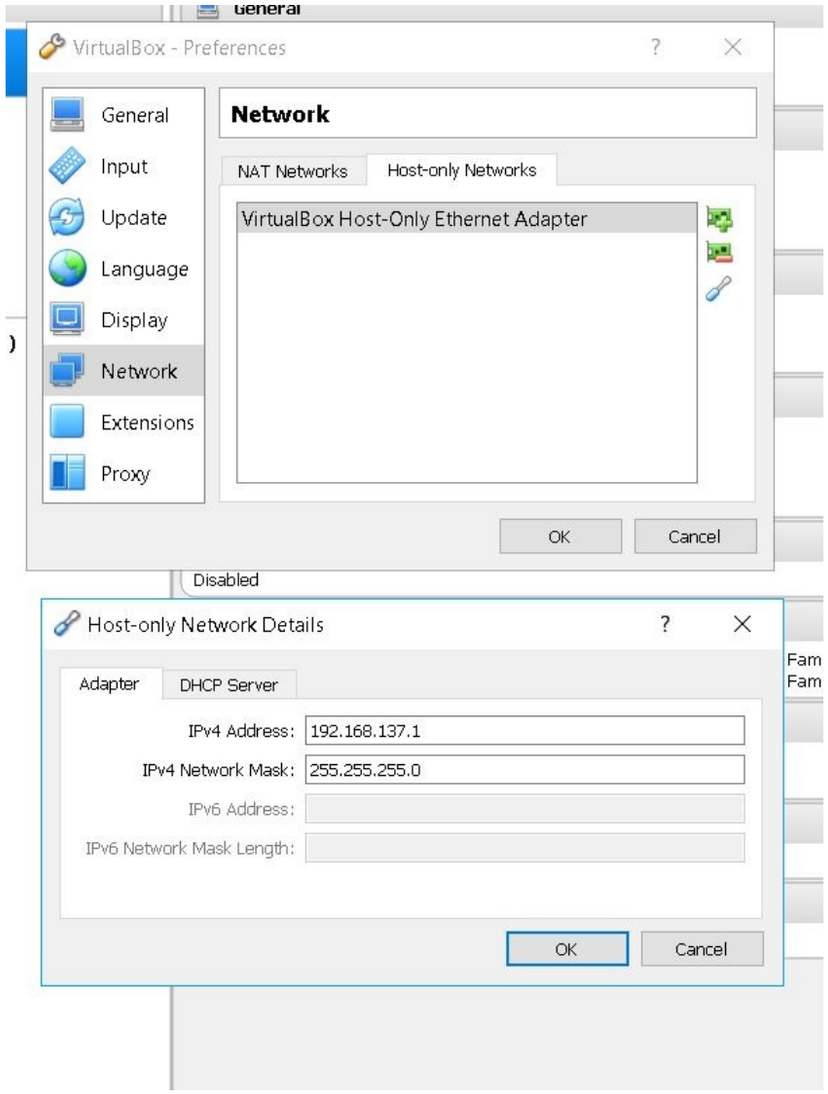

Figure 10.12: 4-Node Network Setup in VirtualBox Configuration-part 3.

10.2 Setup Cloudera Framework and 4-Nodes VM Cluster

The below steps are for installing the Cloudera Manager on Virtual Machines 4-nodes cluster [55].

1. First, we install the Cent-OS Linux 6 and after the installation is complete, we edit the following file:

vi /etc/resolve.conf

Add the following in this file:

- www.domain.com
- nameserver 192.168.1.1

2. We edit the following file:

vi /etc/sysconfig/network

Add the following in this file:

NETWORKING=yes
HOSTNAME=www. n1.com
GATEWAY=192.168.1.1

3. Disable the Selinux to all nodes: vi /etc/selinux/config

SELINUX=disabled

Change the value from enforcing to disabled

4. Run the following command:

chkconfig iptables off

5. Install perl through yum command:

yum –y install perl open ssh-clients

6. Edit our hosts' file:

vi /edit/hosts

Add our hosts in the file (i.e. the below IPs):

- 192.168.1.150 -- www.node1.com
- 192.168.1.151 -- www.node2.com
- 192.168.1.152 -- www.node3.com
- 192.168.1.153 -- www.node4.com

7. Generate ssh-key file by the following command on the master server. Go the following path:

cd /root/.ssh

ssh-keygen

- Those will create two files id_rsa and id_rsa.pub
- Share the id_rsa.pub file at all nodes on which we are installing the Cloudera Manager.

8. Edit the below:

- /etc/ssh/ssh_config
- Change the strictHostkeychecking to No from yes.

9. Update the packages of the system

#yum –y update

10. Apply steps 1 to 9 to all nodes and change IP address and host-names accordingly to the /etc/hosts file

11. Now we download the CDH4 bin installer from the following link on the first server which will act as a base for the installation of the Cloudera Manager.

http://archive.cloudera.com/cm4/installer/4.0.4.14/ cloudera-manager-installer.bin

After that, we transfer the resulting file to the master server, or we can add this file by adding running the following command on the master node of the computer:

curl -0 http://archive.cloudera.com/cm4/installer/latest/cloudera-managerinstaller.bin

12. Now we run the following command to start the installation process:

chmod R 775 Cloudera-manager-installer.bin ./ Cloudera-manager-installer.bin

Assuming that we are in the directory where we have downloaded the binaries. Moreover, we are following the steps as shown in the screenshots.

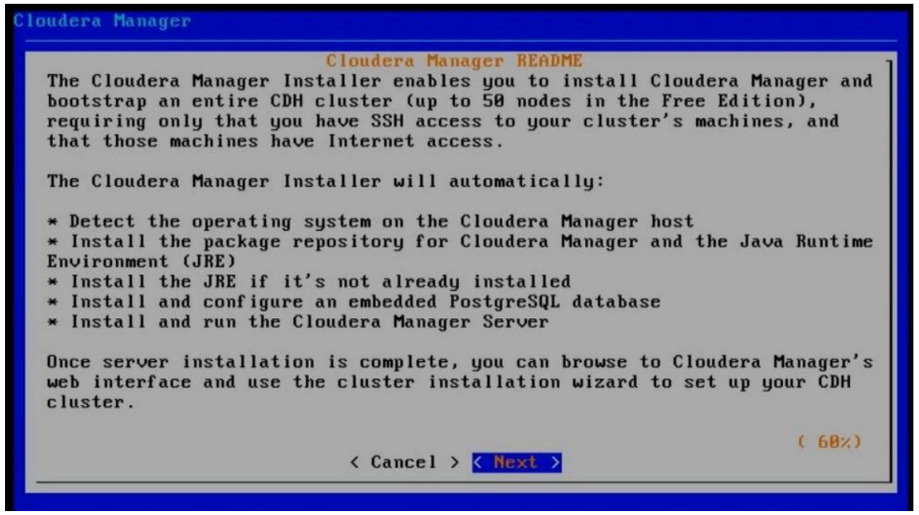

Figure 10.13: Cloudera Manager setup-part 1.

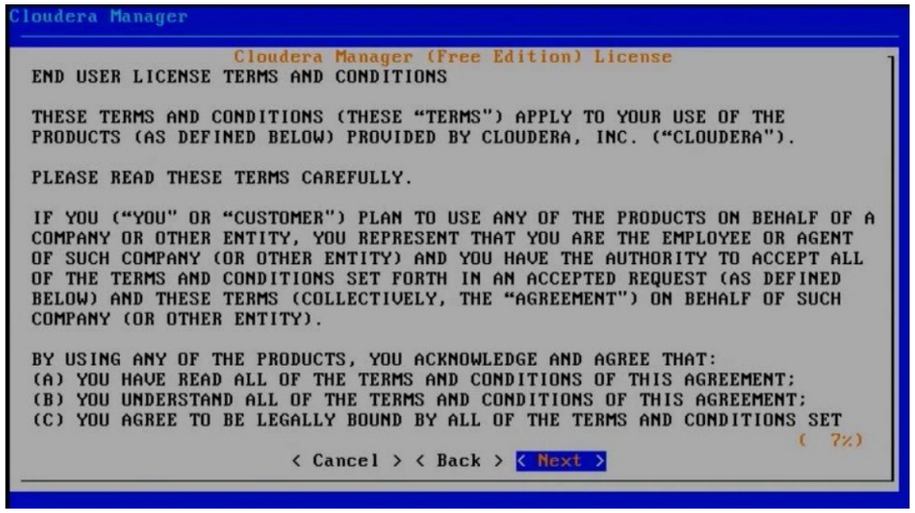

Figure 10.14: Cloudera Manager setup-part 2.

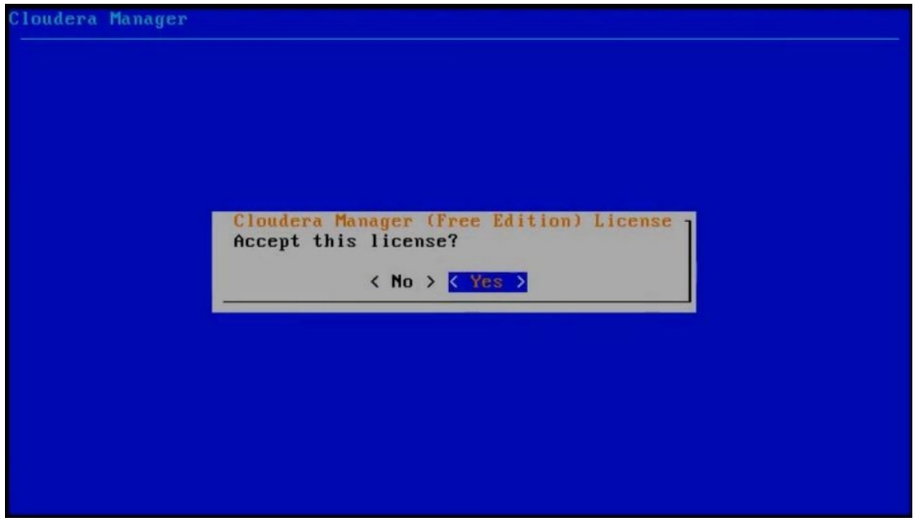

Figure 10.15: Cloudera Manager setup-part 3.

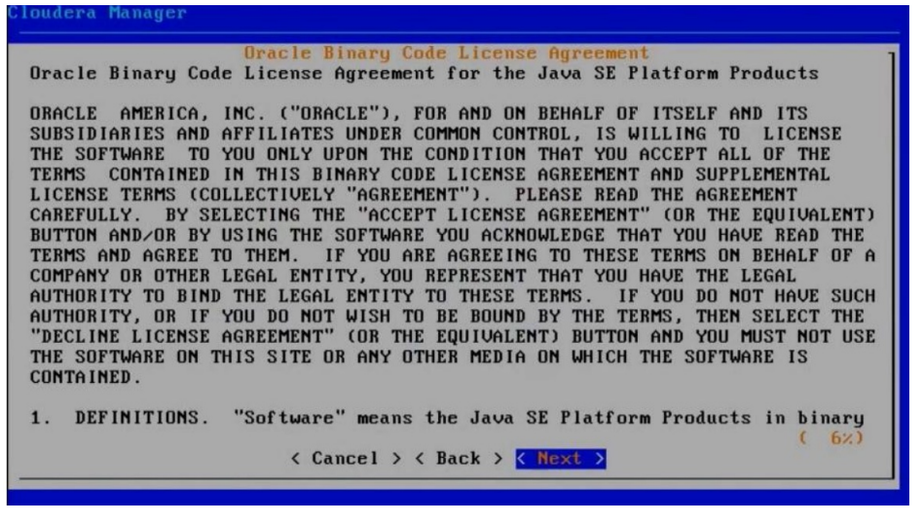

Figure 10.16: Cloudera Manager setup-part 4.

13. After this step our Cloudera Manager has been installed on the base server go to the following link to install and configure the following link:

http://192.168.1.98:7180/

i.e., that is the IP of the base server where we have installed the Cloudera Manager and follow the below screenshots which shows the installation and configuration steps of the Cloudera framework.

14. The standard ID and password are admin, admin respectively.

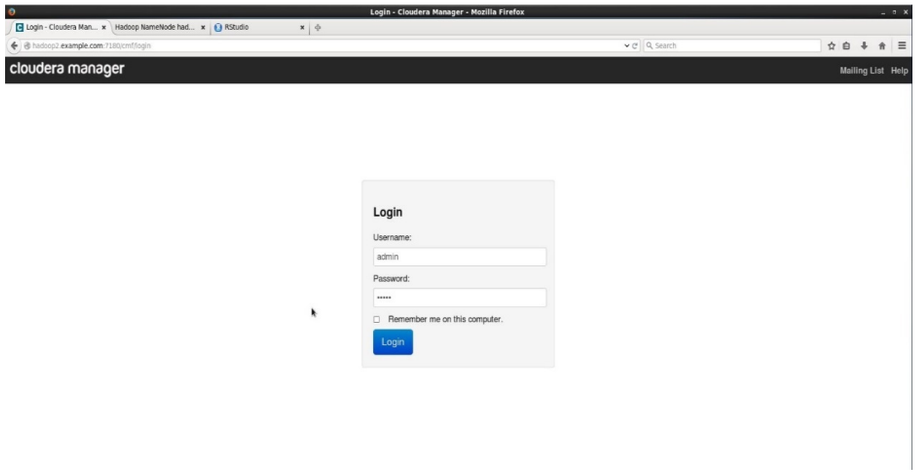

Figure 10.17: Cloudera Manager Login Page.

15. We add our hostnames as defined in your /etc/hosts file on all servers.

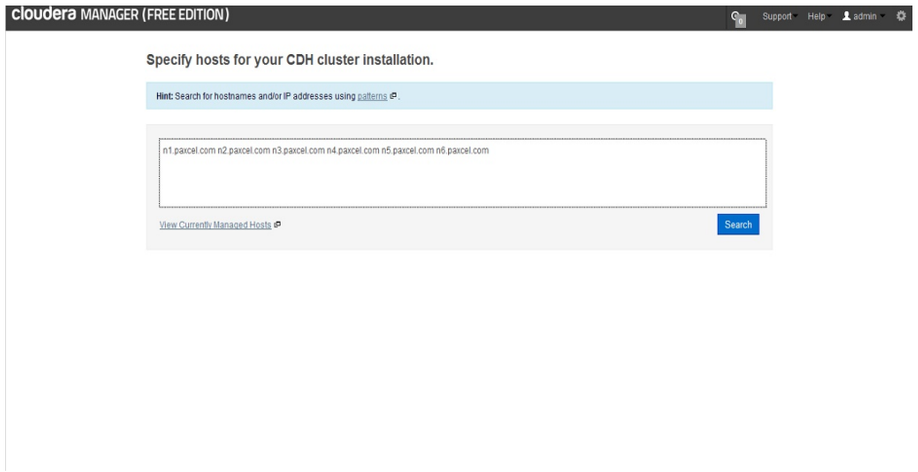

Figure 10.18: Cloudera Manager 4-nodes Cluster setup-part 1.

18. This step is to check the available hosts.
Now we select all the available hosts by checking the text box against it.

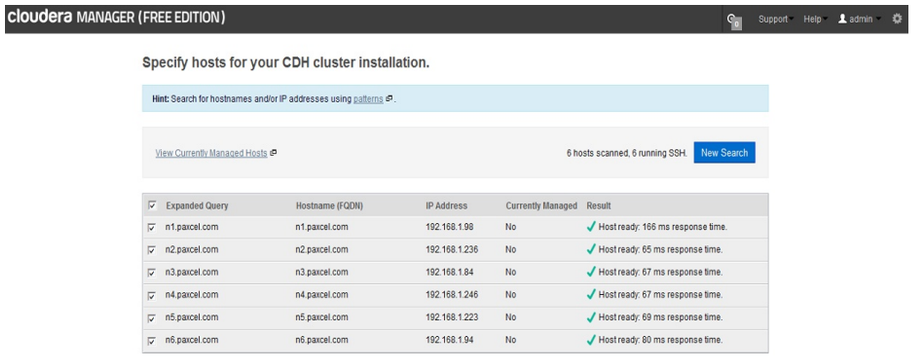

Figure 10.19: Cloudera Manager 4-nodes Cluster setup-part 2.

19. We click on CDH4 Cloudera Manager the latest version of Hadoop.

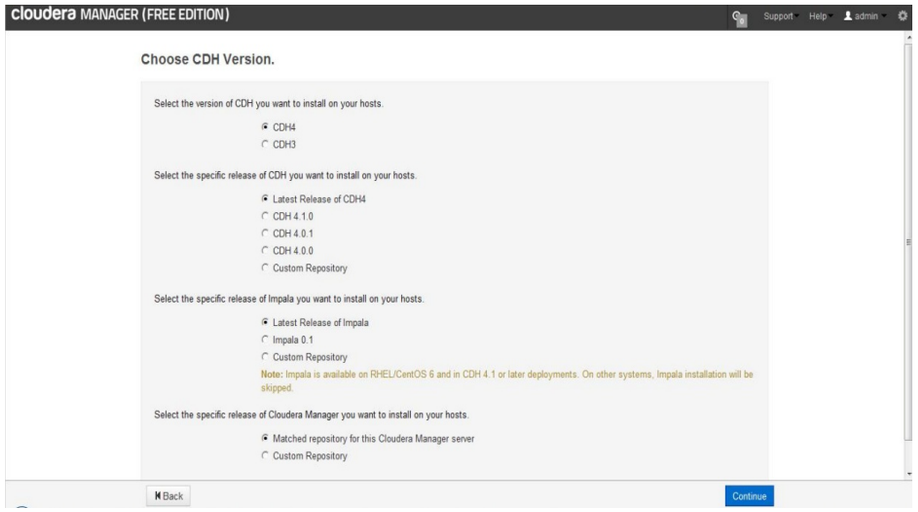

Figure 10.20: Cloudera Manager 4-nodes Cluster setup-part 3.

20. Here either we can upload id_rsa.pub file generated through an ssh-keygen command on the base server and which is shared on all the nodes, or we use same username password on all the nodes.

Here we have chosen same user and password for all the nodes, but it is recommended to go with the encrypted file.

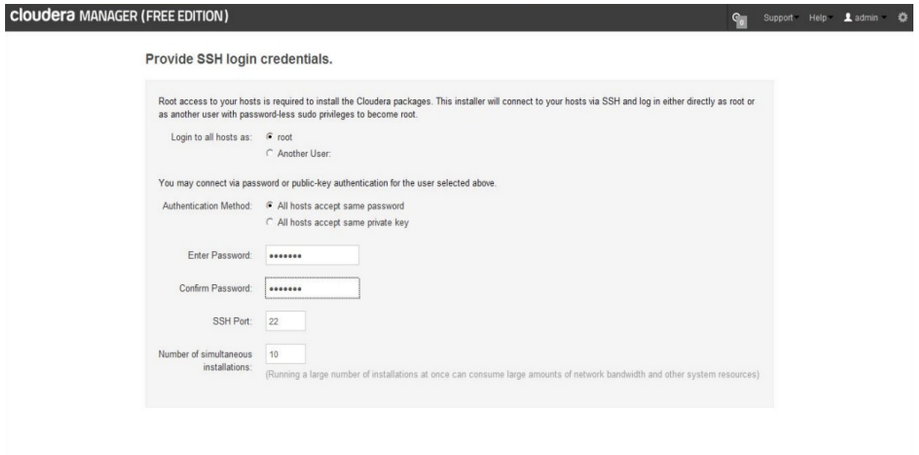

Figure 10.21: Cloudera Manager 4-nodes Cluster setup-part 4.

21. Click on start installation button it will start the installation process on selected six nodes.

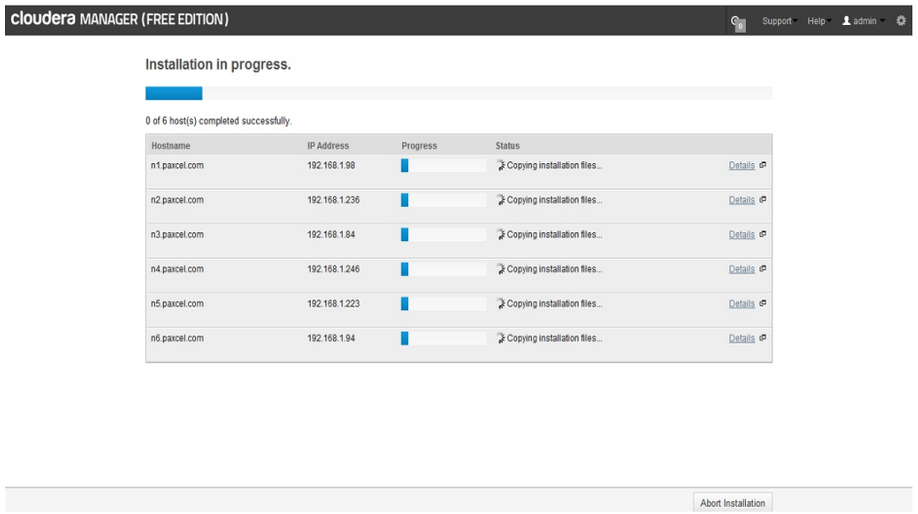

Figure 10.22: Cloudera Manager 4-nodes Cluster setup-part 5.

22. This process takes time for 10 to 15 minutes depending on internet connection speed.

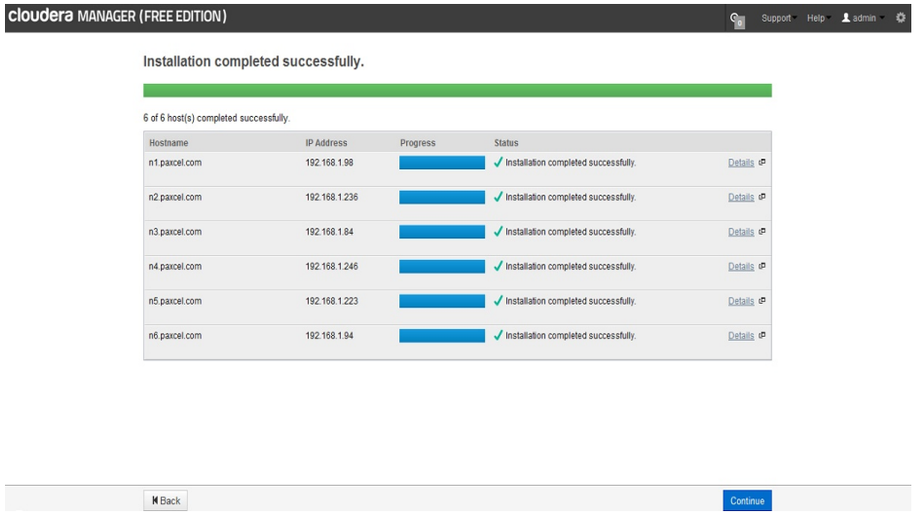

Figure 10.23: Cloudera Manager 4-nodes Cluster setup-part 6.

23. After this process is completed we click on continue button and that will run host inspector to check the host for correctness. After the inspector is finished, we click on continue button.

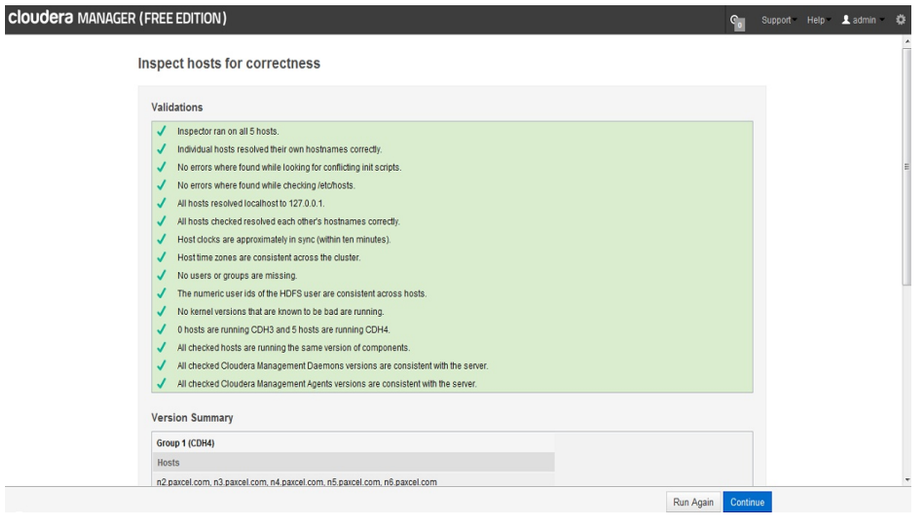

Figure 10.24: Cloudera Manager 4-nodes Cluster setup-part 7.

24. At this stage select CDH4 as shown and it will ask for the services to install. In this case, we have selected the *Core Hadoop* services.

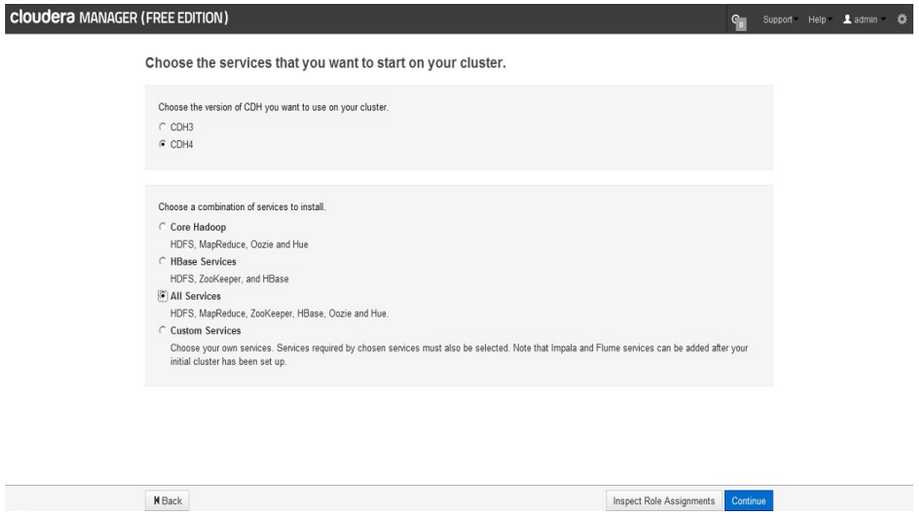

Figure 10.25: Cloudera Manager 4-nodes Cluster setup-part 8.

25. We click on continue option it will configure all services and on the nodes.

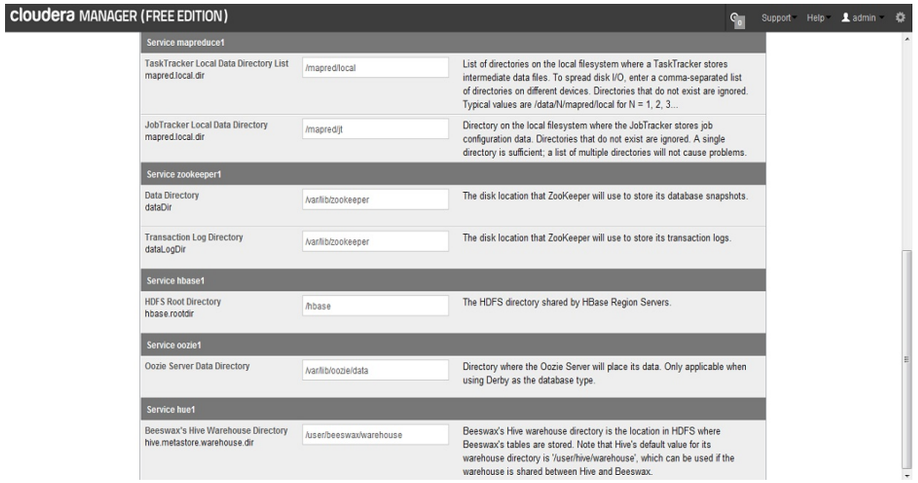

Figure 10.26: Cloudera Manager 4-nodes Cluster setup-part 9.

26. In the below screenshots shows that Cloudera Manager is installing services on the nodes.

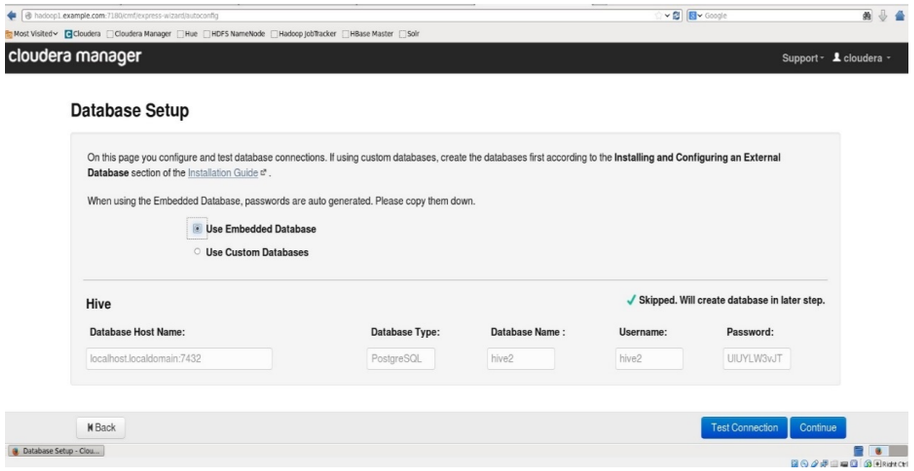

Figure 10.27: Cloudera Manager 4-nodes Cluster setup-part 10.

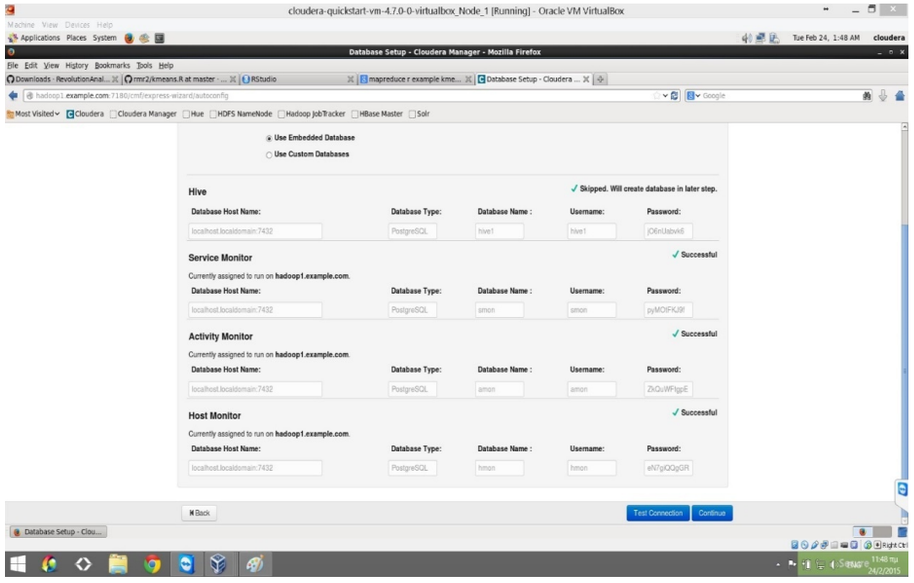

Figure 10.28: Cloudera Manager 4-nodes Cluster setup-part 11.

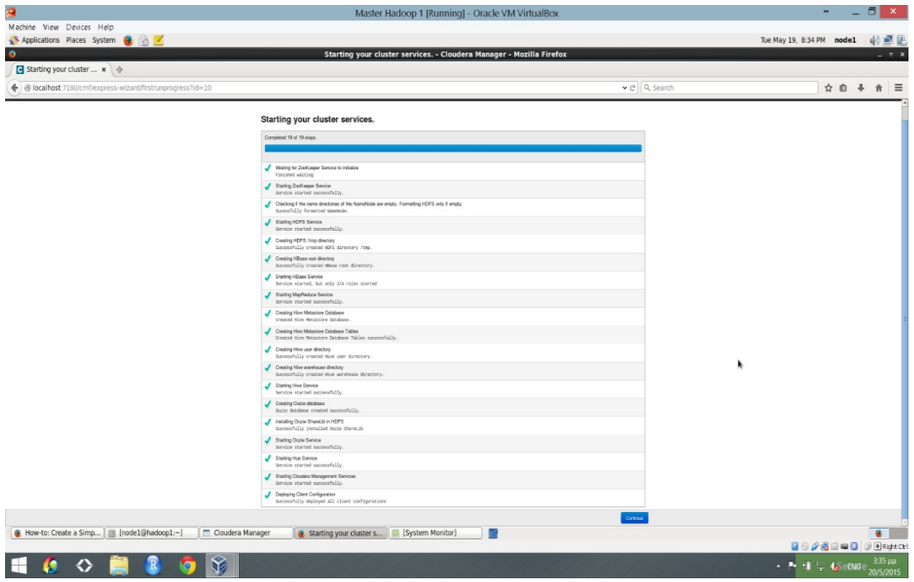

Figure 10.29: Cloudera Manager 4-nodes Cluster setup-part 12.

27. After this, at this point, we are done with installation and configuration of Cloudera Framework.

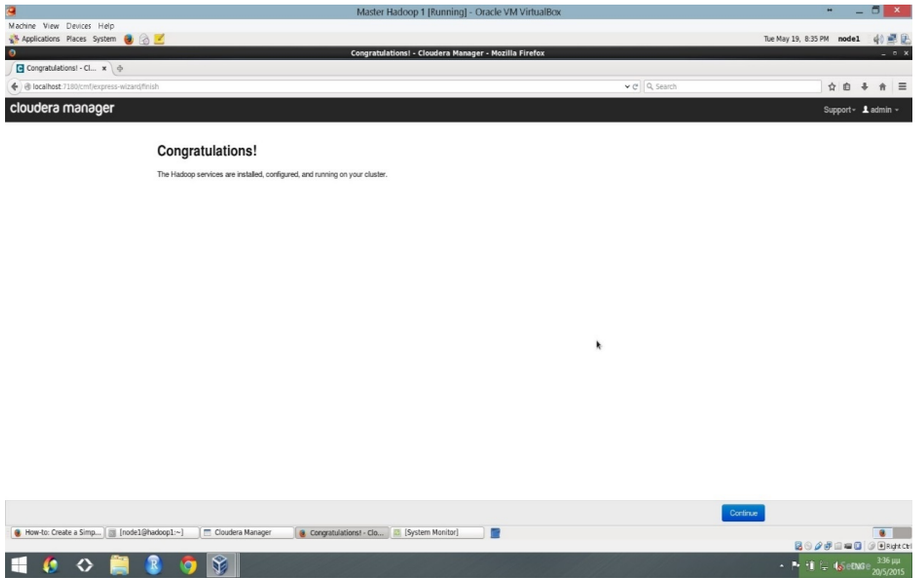

Figure 10.30: Cloudera Manager 4-nodes Cluster setup complete.

28. We login into the console through default id and password, i.e., admin, admin respectively, we can change the password and id also as required.

On the below screenshot shows various services of Hadoop running on the 4-nodes and managed by the Cloudera Manager.

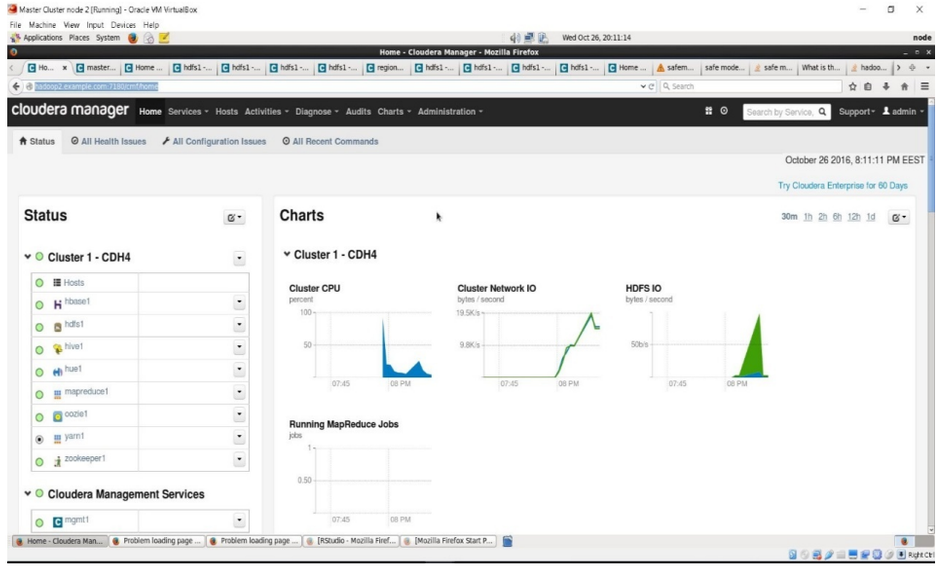

Figure 10.31: Cloudera Manager 4-nodes Cluster Status Monitoring-part 1.

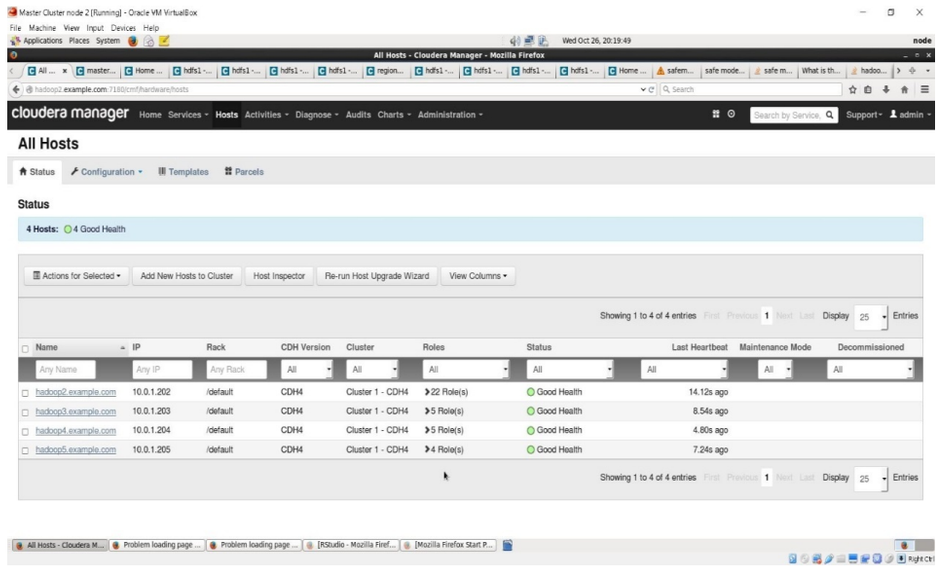

Figure 10.32: Cloudera Manager 4-nodes Cluster Status Monitoring-part 2.

10.3 Installing the R language, the RHadoop and the RStudio

The following is a step-by-step procedure of installing the R language, the RStudio console and the RHadoop components over a Cloudera Hadoop system. The operating system we are using is the Linux CentOS 6.6 [56].

Step 1:

In case we are behind a proxy, as the root user, we set up an environmental variable for the proxy server, as:

export http_proxy=http://192.168.1.254:6588
(http://proxy-server-ip:proxy-serverPort)
And in /etc/yum.conf, write one line as:
proxy=http://192.168.1.254:6588

Recheck our CentOS version, if required:

$ cat /etc/redhat-release

CentOS release 6.6 (Final)

Step 2:

Update our Operating Sytem:

yum -y update

And, we add EPEL repository to the yum configuration:

rpm -ivh http://mirror.chpc.utah.edu/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

(Note that rpmforge has an older version of R. Ee Disable rpmforge, if active by opening file /etc/yum.repos.d/rpmforge.repo and changing the line enabled = 1 to enabled = 0).

Step 3:

Install R & R-devel:

yum install R R-devel

We start an R shell and issue commands within it to download packages as below (note that install.packages() line splits over many lines even though it is one single line till the last closing brackets(s)).

R

(If we are behind proxy, set the proxy server address and port in R-shell as follows before the next command)

> Sys.setenv(http_proxy="http://192.168.1.254:6588″)

If we find strange characters here, the format is:
http://proxy-server:proxyserverPort enclosed within double inverted commas.

> install.packages(c("rJava", "Rcpp", "RJSONIO", "bitops", "digest", "functional", "stringr", "plyr", "reshape2"))
> q()

Step 4:

We download rhdfs and rmr2 packages to our local Download folder from the link:

https://github.com/RevolutionAnalytics/RHadoop/wiki

Step 5:

We set up the environmental variables.
We Write the following three lines in file ~/.bashrc and /etc/profile

export HADOOP_HOME=/usr/lib/hadoop-0.20-mapreduce
export HADOOP_CMD=/usr/bin/hadoop
export HADOOP_STREAMING=/usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.5.0.jar

(Note: Depending upon the Cloudera version we have downloaded, we are checking the version of jar file above and we change it accordingly).

After writing these lines, export the variables as:

# source /etc/profile
$ source ~/.bashrc

Step 6:

We install the downloaded rhdfs & rmr2 packages in the R-shell by specifying its location in our machine.

We start the R:

R

Then, we install the below packages:

> install.packages("/home/ashokharnal/Downloads/rhdfs_1.0.8.tar.gz", repos = NULL, type="source")

> install.packages("/home/ashokharnal/Downloads/rmr2_2.3.0.tar.gz", repos = NULL, type="source")

At this point, R and RHadoop are installed

Step 7:

We Install the RStudio:

```
# wget http://download2.rstudio.org/rstudio-server-0.98.490-x86_64.rpm
# yum install –nogpgcheck rstudio-server-0.98.490-x86_64.rpm
```

Note: RStudio service can be started/stopped as:

```
# service rstudio-server stop
# service rstudio-server status
# service rstudio-server start
```

The RStudio server is accessible at: http://localhost:8787

Step 8:

We restart our machine

Step 9:

We log in as non-root (local) user and we create a file ~/.Rprofile.

We write in it the following seven lines and save the file:

```
# Register the paths
Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for map-reduce
library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()
```

The file ~/.Rprofile contains commands/environmental variables that we need to be set/executed whenever the R starts. This file is picked up by R, as soon as it starts.

Step 10:

Finally, we start the R-shell as a local user and we execute the MR-SSR algorithm as we see below.

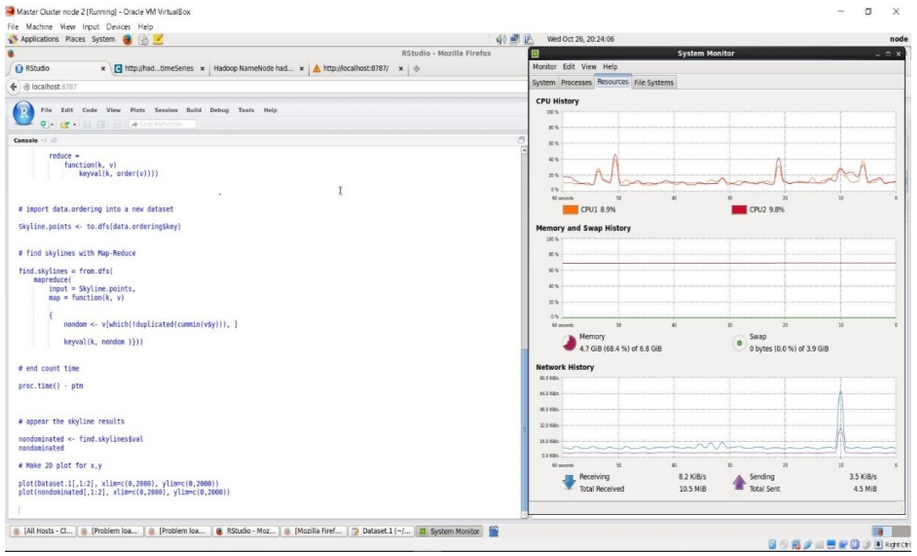

Figure 10.33: Running the Map-Reduce Skyline Algorithm in R Console.

10.4 The Map-Reduce Simple Skyline Algorithm (MR-SSA) in R Language

Below we see, analytically the MR-SSA algorithms which have run in our 4-node computer cluster. These implementations are developed in R language, in two, four, six, eight and ten dimensions with a size of 2000 points of each dimension.

10.4.1 1st implementation: 2 Columns Dataset

```
# Register the paths

Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for map-reduce

library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()

# Create random dataset with 2000 points

x1 = sample.int(2000, 2000)
y1 = sample.int(2000, 2000)

# Choose datasets

Dataset.1 = data.frame(x1,y1)

# Move Dataset to DFS

Dataset.points <- to.dfs(Dataset.1)

 # Start count time

ptm <- proc.time()

# Map-Reduce ordering

data.ordering = from.dfs(
   mapreduce(
      input = Dataset.points,
      map = function(k, v) keyval(v, 1),

            reduce =
          function(k, v)
             keyval(k, order(v))))
```



```
# Import data.ordering into a new dataset

Skyline.points <- to.dfs(data.ordering$key)

# Find skylines with Map-Reduce

find.skylines = from.dfs(
   mapreduce(
      input = Skyline.points,
      map = function(k, v)

      {
         nondom <- v[which(!duplicated(cummin(v$y))), ]

         keyval(k, nondom )}))

# End count time

proc.time() - ptm

# Appear the skyline results

nondominated <- find.skylines$val
nondominated

# Make 2D plot for x1,y1

plot(Dataset.1[,1:2], xlim=c(0,2000), ylim=c(0,2000))
plot(nondominated[,1:2], xlim=c(0,2000), ylim=c(0,2000))
```

10.4.2 2nd implementation: 4 Columns Dataset

```
# Register the paths

Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for map-reduce

library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()
```

```
# Create random dataset with 2000 points

x1 = sample.int(2000, 2000)
y1 = sample.int(2000, 2000)

x2 = sample.int(2000, 2000)
y2 = sample.int(2000, 2000)

# Choose datasets

Dataset.2 = data.frame(x1,y1,x2,y2)

# Move Dataset to DFS

Dataset.points <- to.dfs(Dataset.2)

 # Start count time

ptm <- proc.time()

# Map-Reduce ordering

data.ordering = from.dfs(
   mapreduce(
      input = Dataset.points,
      map = function(k, v) keyval(v, 1),

            reduce =
          function(k, v)
             keyval(k, order(v))))

# Import data.ordering into a new dataset

Skyline.points <- to.dfs(data.ordering$key)

# Find skylines with Map-Reduce

find.skylines = from.dfs(
   mapreduce(
      input = Skyline.points,
      map = function(k, v)

      {
         nondom <- v[which(!duplicated(cummin(v$y))), ]

         keyval(k, nondom )}))

# End count time

proc.time() - ptm
```



```
# Appear the skyline results

nondominated <- find.skylines$val
nondominated

# Make 2D plot for x1,y1,x2,y2

plot(Dataset.2[,1:4], xlim=c(0,2000), ylim=c(0,2000))
plot(nondominated[,1:4], xlim=c(0,2000), ylim=c(0,2000))
```

10.4.3 3rd implementation: 6 Columns Dataset

```
# Register the paths

Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for map-reduce

library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()

# Create random dataset with 2000 points

x1 = sample.int(2000, 2000)
y1 = sample.int(2000, 2000)
z1 = sample.int(2000, 2000)

x2 = sample.int(2000, 2000)
y2 = sample.int(2000, 2000)
z2 = sample.int(2000, 2000)

# Choose  datasets

Dataset.3 = data.frame(x1,y1,z1,x2,y2,z2)

# Move Dataset to DFS

Dataset.points <- to.dfs(Dataset.3)

 # Start count time

ptm <- proc.time()
```

```
# Map-Reduce ordering

data.ordering = from.dfs(
   mapreduce(
      input = Dataset.points,
      map = function(k, v) keyval(v, 1),

            reduce =
           function(k, v)
              keyval(k, order(v))))

# Import data.ordering into a new dataset

Skyline.points <- to.dfs(data.ordering$key)

# Find skylines with Map-Reduce

find.skylines = from.dfs(
   mapreduce(
      input = Skyline.points,
      map = function(k, v)

      {
         nondom <- v[which(!duplicated(cummin(v$y))), ]

         keyval(k, nondom )}))

# End count time

proc.time() - ptm

# Appear the skyline results

nondominated <- find.skylines$val
nondominated

# Make 2D plot for x1,y1,z1,x2,y2,z2

plot(Dataset.3[,1:6], xlim=c(0,2000), ylim=c(0,2000))
plot(nondominated[,1:6], xlim=c(0,2000), ylim=c(0,2000))
```


10.4.4 4th implementation: 8 Columns Dataset

```
# Register the paths

Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for map-reduce

library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()

# Create random dataset with 2000 points

x1 = sample.int(2000, 2000)
y1 = sample.int(2000, 2000)
z1 = sample.int(2000, 2000)

x2 = sample.int(2000, 2000)
y2 = sample.int(2000, 2000)
z2 = sample.int(2000, 2000)

x3 = sample.int(2000, 2000)
y3 = sample.int(2000, 2000)

# Choose datasets

Dataset.4 = data.frame(x1,y1,z1,x2,y2,z2,x3,y3)

# Move Dataset to DFS

Dataset.points <- to.dfs(Dataset.4)

# Start count time

ptm <- proc.time()

# Map-Reduce ordering

data.ordering = from.dfs(
   mapreduce(
      input = Dataset.points,
      map = function(k, v) keyval(v, 1),

            reduce =
           function(k, v)
              keyval(k, order(v))))
```

```
# Import data.ordering into a new dataset

Skyline.points <- to.dfs(data.ordering$key)

# Find skylines with Map-Reduce

find.skylines = from.dfs(
   mapreduce(
      input = Skyline.points,
      map = function(k, v)

      {
         nondom <- v[which(!duplicated(cummin(v$y))), ]

         keyval(k, nondom )}))

# End count time

proc.time() - ptm

# Appear the skyline results

nondominated <- find.skylines$val
nondominated

# Make 2D plot for x1,y1,z1,x2,y2,z2,x3,y3

plot(Dataset.4[,1:8], xlim=c(0,2000), ylim=c(0,2000))
plot(nondominated[,1:8], xlim=c(0,2000), ylim=c(0,2000))
```

10.4.5 5th implementation: 10 Columns Dataset

```
# Register the paths

Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-0.20-mapreduce")
Sys.setenv(HADOOP_CMD="/usr/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh4.7.1.jar")
Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.35.x86_64")

# Load libraries for Map-Reduce

library(rmr2)
library(rJava)
library(rhdfs)
hdfs.init()
```



```
# Create random dataset with 2000 points

x1 = sample.int(2000, 2000)
y1 = sample.int(2000, 2000)
z1 = sample.int(2000, 2000)

x2 = sample.int(2000, 2000)
y2 = sample.int(2000, 2000)
z2 = sample.int(2000, 2000)

x3 = sample.int(2000, 2000)
y3 = sample.int(2000, 2000)
z3 = sample.int(2000, 2000)

x4 = sample.int(2000, 2000)

# Choose datasets

Dataset.5 = data.frame(x1,y1,z1,x2,y2,z2,x3,y3,z3,x4)

# Move Dataset to DFS

Dataset.points <- to.dfs(Dataset.5)

 # Start count time

ptm <- proc.time()

# Map-Reduce ordering

data.ordering = from.dfs(
   mapreduce(
      input = Dataset.points,
      map = function(k, v) keyval(v, 1),

         reduce =
        function(k, v)
          keyval(k, order(v))))

# Import data.ordering into a new dataset

Skyline.points <- to.dfs(data.ordering$key)

# Find skylines with Map-Reduce

find.skylines = from.dfs(
   mapreduce(
      input = Skyline.points,
      map = function(k, v)

      {
```

```
        nondom <- v[which(!duplicated(cummin(v$y))), ]

        keyval(k, nondom )}))
```

 # End count time

```
proc.time() - ptm
```

 # Appear the skyline results

```
nondominated <- find.skylines$val
nondominated
```

 # Make 2D plot for x1,y1,z1,x2,y2,z2,x3,y3,z3,x4

```
plot(Dataset.5[,1:10], xlim=c(0,2000), ylim=c(0,2000))
plot(nondominated[,1:10], xlim=c(0,2000), ylim=c(0,2000))
```


11. References

[1] A. Drigas and P. Leliopoulos, "Business to Consumer (B2C) E-Commerce Decade Evolution," *Int. J. Knowl. Soc. Res. IJKSR*, vol. 4, no. 4, pp. 1–10, Dec. 2013.

[2] A. S. Drigas and P. Leliopoulos, "The Use of Big Data in Education," *Int. J. Comput. Sci. Issues IJCSI*, vol. 11, no. 5, pp. 58 – 63, Sep. 2014.

[3] C. Tsaravas and M. Themistocleous, "CLOUD COMPUTING & EGovernment: MYTH OR REALITY?"

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[5] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, no. 3, p. 37, 1996.

[6] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.

[7] P. Whitla, "Crowdsourcing and its application in marketing activities," *Contemp. Manag. Res.*, vol. 5, no. 1, 2009.

[8] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *Syst. Man Cybern. IEEE Trans. On*, vol. 24, no. 4, pp. 656–667, 1994.

[9] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, no. 2, pp. 95–99, 1988.

[10] G. G. Chowdhury, "Natural language processing," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, 2003.

[11] V. Batagelj and A. Mrvar, "Pajek-program for large network analysis," *Connections*, vol. 21, no. 2, pp. 47–57, 1998.

[12] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008.

[13] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Process.*, vol. 80, no. 10, pp. 2219–2235, 2000.

[14] S. Chaudhuri, "What next?: a half-dozen data management research goals for big data and the cloud," in *Proceedings of the 31st symposium on Principles of Database Systems*, 2012, pp. 1–4.

[15] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu, "Starfish: A self-tuning system for big data analytics," in *Proc. of the Fifth CIDR Conf*, 2011.

[16] D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, and J. Han, "Challenges and Opportunities with Big Data 2012-2," 2011.

[17] J. Bughin, M. Chui, and J. Manyika, "Clouds, big data, and smart assets: Ten tech-enabled business trends to watch," *McKinsey Q.*, vol. 56, 2010.

[18] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, and A. Rasin, "HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 922–933, 2009.

[19] D. Bollier and C. M. Firestone, *The promise and peril of big data*. Aspen Institute, Communications and Society Program, 2010.

[20] A. Bhatia and G. Vaswani, "BIG Data–A Review."

[21] M. Themistocleous, K. Koumaditis, and G. Vassilacopoulos, "PROVIDING INTEGRATED E-HEALTH SERVICES FOR PERSONALIZED MEDICINE UTILIZING CLOUD INFRASTRUCTURE."

[22] S. Guthe, M. Wand, J. Gonser, and W. Straßer, "Interactive rendering of large volume data sets," in *Visualization, 2002. VIS 2002. IEEE*, 2002, pp. 53–60.

[23] B. Brown, M. Chui, and J. Manyika, “Are you ready for the era of ‘big data’?,” *McKinsey Q.*, vol. 4, pp. 24–35, 2011.
[24] H. Chen, R. H. Chiang, and V. C. Storey, “Business Intelligence and Analytics: From Big Data to Big Impact.,” *MIS Q.*, vol. 36, no. 4, pp. 1165–1188, 2012.
[25] O. Tigas, P. Psathopoulou, C. Radulescu, and P. Leliopoulos, “The use of MS Expression Blend for generating interactive applications in University education,” in *Exploit Technologies Information and Communication Teaching Practice*, Syros, 2013.
[26] J. Bilbao, G. Garate, A. Olozaga, and A. del Portillo, “Easy clustering with openMosix,” in *Proceedings of the 9th WSEAS International Conference on Computers*, 2005, pp. 1–6.
[27] M. Bakery and R. Buyyaz, “Cluster computing at a glance,” *High Perform. Clust. Comput. Archit. Syst. Up. Saddle River NJ Prentice-Hall*, pp. 3–47, 1999.
[28] C. Leangsuksun, L. Shen, T. Liu, H. Song, and S. L. Scott, “Availability prediction and modeling of high mobility OSCAR cluster,” in *Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on*, 2003, pp. 380–386.
[29] P. M. Papadopoulos, M. J. Katz, and G. Bruno, “NPACI Rocks: Tools and techniques for easily deploying manageable linux clusters,” *Concurr. Comput. Pract. Exp.*, vol. 15, no. 7–8, pp. 707–725, 2003.
[30] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, “A high-performance, portable implementation of the MPI message passing interface standard,” *Parallel Comput.*, vol. 22, no. 6, pp. 789–828, 1996.
[31] C. Tian, H. Zhou, Y. He, and L. Zha, “A dynamic mapreduce scheduler for heterogeneous workloads,” in *Grid and Cooperative Computing, 2009. GCC’09. Eighth International Conference on*, 2009, pp. 218–224.
[32] “Hadoop Tutorial: Hello World - An Overview of Hadoop with HCatalog, Hive and Pig,” *Hortonworks*.
[33] M. Olson, “Hadoop: Scalable, flexible data storage and analysis,” *IQT Q.*, vol. 1, no. 3, pp. 14–18, 2010.
[34] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[35] I. Aljarah and S. A. Ludwig, “Parallel particle swarm optimization clustering algorithm based on MapReduce methodology,” in *Nature and Biologically Inspired Computing (NaBIC), 2012 Fourth World Congress on*, 2012, pp. 104–111.
[36] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis, “Evaluating mapreduce for multi-core and multiprocessor systems,” in *High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on*, 2007, pp. 13–24.
[37] F. Marozzo, D. Talia, and P. Trunfio, “Adapting MapReduce for dynamic environments using a peer-to-peer model,” in *Proceedings of the 1st Workshop on Cloud Computing and its Applications*, 2008.
[38] J. Ekanayake, S. Pallickara, and G. Fox, “Mapreduce for data intensive scientific analyses,” in *eScience, 2008. eScience’08. IEEE Fourth International Conference on*, 2008, pp. 277–284.
[39] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, “Twister: a runtime for iterative mapreduce,” in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, 2010, pp. 810–818.
[40] J. Dittrich and J.-A. Quiané-Ruiz, “Efficient big data processing in Hadoop MapReduce,” *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 2014–2015, 2012.
[41] D. Borthakur, J. Gray, J. S. Sarma, K. Muthukkaruppan, N. Spiegelberg, H. Kuang, K. Ranganathan, D. Molkov, A. Menon, and S. Rash, “Apache Hadoop goes realtime at Facebook,” in *Proceedings of the 2011 international conference on Management of data*, 2011, pp. 1071–1080.
[42] “Hadoop Distributed File System (HDFS),” *Hortonworks*.
[43] S. Sharma, “Big Data Landscape.”
[44] R. C. Taylor, “An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics,” *BMC Bioinformatics*, vol. 11, no. Suppl 12, p. S1, 2010.
[45] Cloudera, “Frequently Asked Questions (FAQs) (n.d.).”.
[46] Hewlett-Packard Development Company, “HP Reference Architecture for Hortonworks Data Platform on HP ProLiant SL4540 Gen8 Server,” 2013.
[47] A. Georgiou, “Storing Data Flow Monitoring in Hadoop,” 2013.
[48] BigHadoop, "R and Hadoop Data Analysis – RHadoop," https://bighadoop.wordpress.com/2013/02/25/r-and-hadoop-data-analysis-rhadoop, 2013.
[49] P. Leliopoulos, “Application development in cloud computing: calculation Skyline points based on MapReduce method using the R language,” http://dione.lib.unipi.gr/xmlui/handle/unipi/8942, 2015.
[50] RevolutionAnalytics,“RHadoop,” ttps://github.com/RevolutionAnalytics/RHadoop/wiki, 2015.
[51] Mullesgaard, Kasper et al. "Efficient Skyline Computation in MapReduce". Proc. 17th International Conference on Extending Database Technology (EDBT). 2014: 37-48.
[52] Liang Chen, Kai Hwang, Jian Wu: MapReduce Skyline Query Processing with a New Angular Partitioning Approach. IPDPS Workshops 2012: 2262-2270.
[53] Zhang, Boliang, Shuigeng Zhou, and Jihong Guan. "Adapting Skyline Computation to the MapReduce Framework: Algorithms and Experiments." DASFAA Workshops. 2011: 403-414.
[54] Wikipedia, "Multi-objective optimization", https://en.wikipedia.org/wiki/Multi-objective_optimization#Examples_of_multi-objective_optimization_applications, August 2017.
[55] Avinash K. Singh, “Hadoop Introduction, Installation and Administration”, https://docs.google.com/file/d/0Bx6N95pJhrRObljiaEJ0dHpwVmc/edit, 2012.
[56] Linux Uncle, “Installing R, RHadoop and RStudio over Cloudera Hadoop ecosystem (Revised)”, https://ashokharnal.wordpress.com/2014/01/16/installing-r-rhadoop-and-rstudio-over-cloudera-hadoop-ecosystem-revised/, 2014.

Proof