IMPROVING THE PROCESSING OF LARGE VOLUME OF DATA USING HADOOP

A Thesis Submitted
In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF TECHNOLOGY
In
Field of Specialization

By
Suyogita Singh
(Enrollment no: 11804490803)
Under the Supervision of
Associate Professor
Mr. Sameer Awasthi
CSE Department BBDU, Lucknow

BBD UNIVERSITY

to the
SCHOOL OF ENGINEEGING

BABU BANARASI DAS UNIVERSITY
LUCKNOW

JUNE, 2020

<u>CERTIFICATE</u>

It is certified that the work contained in this thesis entitled **“improving the processing of large volume of data using HADOOP”** by Suyogita Singh (Roll N0.1180449005) for the award of **Master of Technology** from Babu Banarasi Das University has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Mr. Sameer Awasthi

(Associate Professor)

Babu Banarasi Das University

Date:

ABSTRACT

Today, we’re surrounded by data like oxygen. The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, Microsoft, Facebook, Twitter etc. Data volumes to be processed by cloud applications are growing much faster than computing power. This growth demands new strategies for processing and analyzing information. Hadoop-Map Reduce has become a powerful Computation Model addresses to these problems. Hadoop HDFS became more popular amongst all the Big Data tools as it is open source with flexible scalability, less total cost of ownership & allows data stores of any form without the need to have data types or schemas defined. Hadoop Map Reduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes. In this paper I have provided an overview, architecture and components of Hadoop, HCFS (Hadoop Cluster File System) and Map Reduce programming model, its various applications and implementations in Cloud Environments.

ACKNOWLEDGMENT

It gives me blissfulness in communicating my significant feeling of appreciation to my thesis supervisor Mr. Sameer Awasthi. I am heavily grateful and profoundly appreciative to him for his agitate direction, consistent consolation, important proposals and quick thoughts. His colossal information and comprehension has significantly helped me in better comprehension of different parts of Biometrics. His skill in the zone of Biometrics and excitement in the exploration have been of incredible incentive to me.

We are likewise appreciative to Mr. Sameer Awasthi, Dr. Praveen Kumar Shukla and other in Department of Computer Science and Engineering for rousing us in improving the calculations.

At long last we might want to thank our folks for their help and allowing us remains for additional days to finish this task.

Suyogita Singh
Babu Banarsi Das University

LIST OF TABLES

Page No.

LIST OF FIGURES

LIST OF SYSMBOLS AND ABBREVIATIONS

AOSD	Aspect-Oriented Software Development
AQL	Annotation Query Language
BDA	Big Data Analytics
EPN	Event Processing Nodes
ETL	Extract, transform and load
FUSE	File system in User space
HAR	Hadoop Archive
HPCC	High Performance Computing Cluster
HPDL	Hadoop Process Definition Language
HPIL	Hadoop Physical Infrastructure Layer
JSON	JavaScript Object Notation Query Language

TABLE OF CONTENT

Chapter 1

Introduction

The IT industry is always developing new technologies, and big data is the one of them. With the developments of the cloud storage, big data has attracted more and more attention. Due to the emergence of the Internet, the big data technology will accelerate the innovation of the enterprises, lead the revolution of the business mode and create unlimited commercial opportunities. Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software. Data with many cases (rows) offer greater statistical power, while data with higher complexity (more attributes or columns) may lead to a higher false discovery rate. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source. Big data was originally associated with three key concepts volume, variety, and velocity. When we handle big data, we may not sample but simply observe and track what happens. Therefore, big data often includes data with sizes that exceed the capacity of traditional software to process within an acceptable time and value. Current usage of the term big data tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. "There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem." Analysis of data sets can find new correlations to "spot business trends, prevent diseases, and combat crime and so on." Scientists, business executives, practitioners of medicine, advertising and governments alike regularly meet difficulties with large data-sets in areas including Internet searches, fintech, urban informatics, and business informatics.

1.1 Current situation of the big data

In recent years, we have been drowning in the ocean of data that was produced by the development of the Internet, the Mobile Internet, the Internet of Things (loT) and the Social Networks. A photo that uploaded to Instagram is about 1MB; a video that uploaded to YouTube is about dozens of Mega sizes. Chatting online, browsing websites, playing online games, and shopping online will also turn into data that may be stored in any corner in the world. Hence, how much data is there in our daily life? According to a report of IBM, there are 2.5 quintillion bytes of data that we create every day. Ninety percent of that data was created in the recent two years. That means, in one day, the data that appears on the Internet can fill 168 million DVDs the volume of data has increased from Terabyte level to Petabyte level. The reducing price of computer hardware and the production of supercomputers make it possible to deal with large and complex data.

1.2 The definition of Big Data

Finding a way to define the Big Data is not easy, but authors hold the same view with Ohlhosrt (2012) that Big Data is the large and complex data that is difficult to use the traditional tools to store, manage, and analyze in an acceptable duration. Therefore, the Big Data needs a new processing model which has the better storage, decision-making, and analyzing abilities. This is the reason why the Big Data technology was born. The Big Data technology provides a new way to extract, interact, integrate, and analyze of Big Data. The Big Data strategy is aiming at mining the significant valuable data information behind the Big Data by specialized processing. In other words, if comparing the Big Data to an industry, the key of the industry is to create the data value by increasing the processing capacity of the data.

1.3 Types of Big Data

1.3.1 Structured Data

Any data that can be stored, accessed and processed in the form of fixed format is termed as 'structured' data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the rage of multiple petabytes.

1.3.2 Unstructured

Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, un-structured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format.

1.3.3 Semi-structured

Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in an XML file.

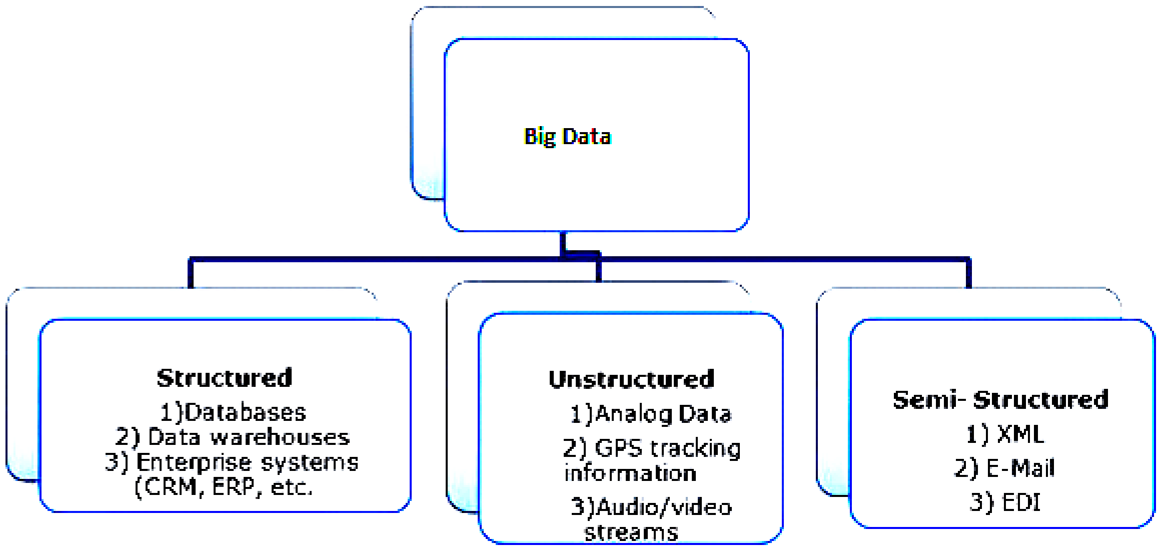

Figure 1.1 Different types of data

1.4 The characteristics of Big Data

According to a research report from Gartner (Doug, 2001), the growth of the data is three-dimensional, which is volume, velocity and variety. So far, there are many industries still use the 3Vs model to describe the Big Data. However, the Big Data is not only 3Vs but also has some other characteristics. The first one is the volume. As mentioned, the volume of Big Data has moved from Terabyte level to Petabyte level. The second one is the variety. Compared with the traditional easy to storage structured text data, there is now an increasing amount unstructured data that contains web logs, audio, video, pictures, and locations. Data no longer needs to be stored as traditional tables in databases or data warehouses but also stored as variable data types at the same time. To meet this requirement, it is urgent to improve the data storage abilities. Next is velocity. Velocity is the most significant feature to distinguish the Big Data and the traditional data mining. In the Age of Big Data, the volume of high concurrency access of users and submittion data are huge. Velocity of interactive response occurs when a user submits his or her request to the server and the corresponding speed should be fast instead of making the users waiting for a long time. Taking the Facebook as example, when billions of users submit their request to access to Facebook at the same time, this requires a rapid speed of interactive response to each user. The rate of valuable data is in inverse proportion to the total data volume so that the ratio of valuable data is quite low. For instance, under the continuously monitoring of a one-hour video, the useful data may be a few seconds only. How to combine the business logic and powerful algorithm to mine the treasure is the hardest puzzle of the big data technology. Last but not at least is the online property. Big Data is always online and can be accessed and computed. With the rapid developments of the Internet, the Big Data is not only big but is also getting online. Online data is meaningful when the data connects to the end users or the customers. Taking an example, when users use Internet applications, the users' behavior will be delivered to the developers immediately. These developers will optimize the notifications of the applications by using some methods to analyze the data. Pushing the notifications that users want to see most is also a brilliant way to promote the user experience.

Volume: With all the data in the world growing two times in every two years, there is a need for an efficient and scalable platform to serve the large datasets that are going to be produced in future as well.

Variety: The database system should be able to store and process of a variety of structured, unstructured or semi-structured data in any format (Examples: XML, JSON, CSV, JPEG, PDF).

Veracity: Gone are the days, where few business executives take market decisions. The truthfulness of the decisions made by analytics depends upon the veracity of the data, which ultimately drives business.

Velocity: The speed at which data is growing is tremendous and there lies a need for organizations to predict future's data.

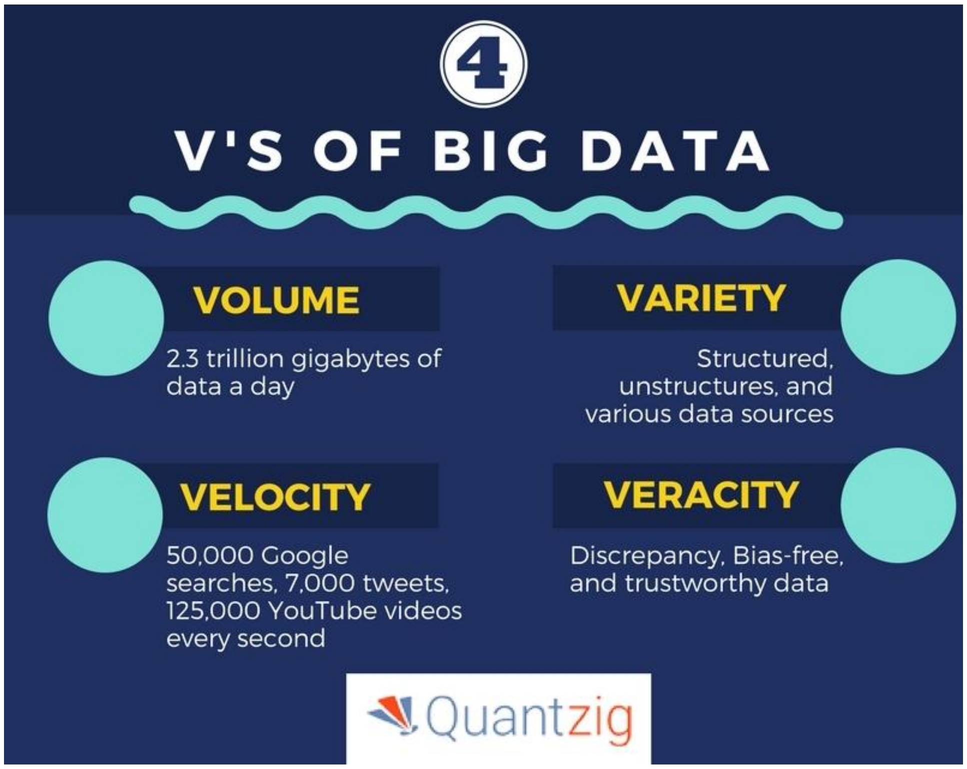
4
V'S OF BIG DATA
VOLUME
2.3 trillion gigabytes of data a day
VARIETY
Structured, unstructures, and various data sources
VELOCITY
50,000 Google searches, 7,000 tweets, 125,000 YouTube videos every second
VERACITY
Discrepancy, Bias-free, and trustworthy data
Quantzig

Figure 1.2 4V'S of big data

1.4.1 Other important characteristics of Big Data are

- Exhaustive: Whether the entire system (i.e., =all) is captured or recorded or not.
- Fine-grained and uniquely lexical: Respectively, the proportion of specific data of each element per element collected and if the element and its characteristics are properly indexed or identified.
- Relational: If the data collected contains common fields that would enable a conjoining, or meta-analysis, of different data sets.
- Extensional: If new fields in each element of the data collected can be added or changed easily.
- Scalability: If the size of the data can expand rapidly.
- Value: The utility that can be extracted from the data.
- Variability: It refers to data whose value or other characteristics are shifting in relation to the context in which they are being generated.

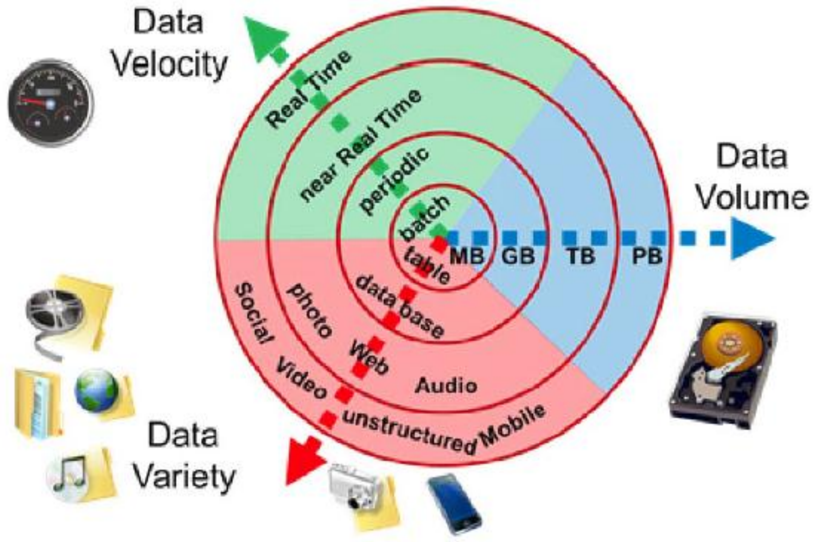

Figure 1.3 shows the growth of big data

1.5 Architecture of big Data

Big data repositories have existed in many forms, often built by corporations with a special need. Commercial vendors historically offered parallel database management systems for big data beginning in the 1990s. For many years, Intercrop published the largest database report.

Teradata Corporation in marketed the parallel processing DBCS 1012 system. Teradata systems were the first to store and analyze 1 terabyte of data in 1992. Hard disk drives were 2.5 GB in 1991 so the definition of big data continuously evolves according to Kryder's Law. Teradata installed the first petabyte class RDBMS based system in 2007. As of 2017, there are a few dozen petabyte class Teradata relational databases installed, the largest of which exceeds 50 PB. Systems up until 2008 were 100% structured relational data. Since then, Teradata has added unstructured data types including XML.

In 2000, Seisint Inc. developed a C++-based distributed platform for data processing and querying known as the HPCC Systems platform. This system automatically partitions, distributes, stores and delivers structured, semi-structured, and unstructured data across multiple commodity servers. Users can write data processing pipelines and queries in a declarative dataflow programming language called ECL.

Data analysts working in ECL are not required to define data schemas upfront and can rather focus on the particular problem at hand, reshaping data in the best possible manner as they develop the solution. In 2004, LexisNexis acquired Seisint Inc. and their high-speed parallel processing platform and successfully utilized this platform to integrate the data systems of Choice point Inc. when they acquired that company in 2008. In 2011, the HPCC systems platform was open-sourced under the Apache v2.0 License.CERN and other physics experiments have collected big data sets for many decades, usually analyzed via high-throughput computing rather than the map-reduce architectures usually meant by the current "big data" movement.In 2004, Google published a paper on a process called Map Reduce that uses a similar architecture. The Map Reduce concept provides a parallel processing model, and an associated implementation was released to process huge amounts of data. With Map Reduce, queries are split and distributed across parallel nodes and processed in parallel (the Map step). The results are then gathered and delivered (the Reduce step). The framework was very successful, so others wanted to replicate the algorithm. Therefore, an implementation of the Map Reduce framework was adopted by an Apache open-source project named Hadoop. Apache Spark was developed in 2012 in response to limitations in the Map Reduce paradigm, as it adds the ability to set up many operations (not just map followed by reducing).MIKE2.0 is an open approach to information management that acknowledges the need for revisions due to big data implications identified in an article titled "Big Data Solution Offering". The methodology addresses handling big data in terms of useful permutations of data sources, complexity in interrelationships, and difficulty in deleting (or modifying) individual records. 2012 studies showed that multiple-layer architecture is one option to address the issues that big data presents. A distributed parallel architecture distributes data across multiple servers; these parallel execution environments can dramatically improve data processing speeds. This type of architecture inserts data into a parallel DBMS, which implements the use of Map Reduce and Hadoop frameworks. This type of framework looks to make the processing power transparent to the end-user by using a front-end application server. The data lake allows an organization to shift its focus from centralized control to a shared model to respond to the changing dynamics of information management. This enables quick segregation of data into the data lake, thereby reducing the overhead time.

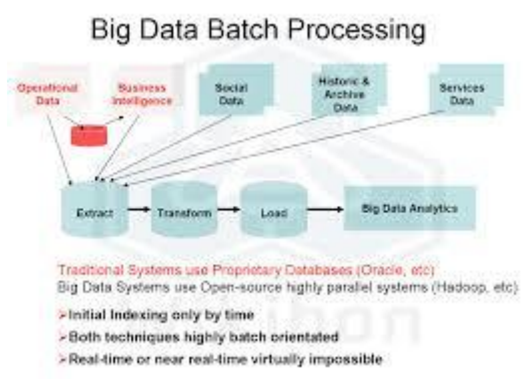
Big Data Batch Processing
Operational Data
Business Intelligence
Social Data
Historic & Archive Data
Services Data
Extract
Transform
Load
Big Data Analytics
Traditional Systems use Proprietary Databases (Oracle, etc)
Big Data Systems use Open-source highly parallel systems (Hadoop, etc)
➢Initial Indexing only by time
➢Both techniques highly batch orientated
➢Real-time or near real-time virtually impossible

Figure 1.4 Big Data Processing

1.6 Basic Data Processing Platform

Distributed Data Processing (DDP) is not only a technical concept but also a logical structure. The concept of DDP is based on the principle that can achieve both centralized and decentralized information service.

1.6.1 Capability components of DDP Platform

DDP platforms have different capability components to help it to complete the whole process. Different capability components are responsible for different jobs and aspects. The following sections will introduce the most important capability components of a DDP platform.

- **File Storage:** The file storage capability component is the basic unit of data management in the data processing architecture. It aims to provide a fast and reliable access ability to meet the needs of large amount of data computing.

- **Data Storage:** The data storage capability component is an advanced unit of data management in the data processing architecture. It aims to store the data according to an organized data model and to provide an independent ability of deleting and modifying data. IBM DB2 is a good example of a data storage capability component.

- **Data Integration:** The data integration capability component integrates the different data which has different sources, formats, and characters into units to support the data input and output between multiple data sources and databases. Oracle Data Integrator is an example of data integration component.
- **Data Computing:** The data computing capability component is the core component of the whole platform. It aims to solve the specific problem by using the computing resources of the processing platform. Taking MPI (Message Passing Interface) which is

commonly used in parallel computing as an example, it is a typical data computing component. In the Big Data environment, the core problem is how to split the task that needs huge computing ability to calculate.

- **Data analysis:** The data analysis capability component is the closest component to the users in the data processing platform. It aims to provide an easy way to support the user to extract the data related to their purpose from the complex information. For instance, as a data analysis component, SQL (Structured Query Language) provides a good analysis method for the relational databases. Data analysis aims at blocking the complex technical details in the bottom layer of the processing platform for the users by abstract data access and analysis. Through the coordinates of data analysis components, the users can do the analysis by using the friendly interfaces rather than concentrate on data storage format, data streaming and file storage.

- **Platform Management:** The platform management capability component is the managing component of the data processing. It aims to guarantee the safety and stability for the data processing. In the Big Data processing platform, it may consist of a large amount of servers that may be distributed in different locations. On this occasion, how to manage these servers' work efficiently to ensure the entire system running is a tremendous challenge.

1.7 Data processing Using Big Data

Big Data processing involves steps very similar to processing data in the transactional or data warehouse environments. Figure 1.5 shows the different stages involved in the processing of Big Data; the approach to processing Big Data is The connection between big data and data preprocessing throughout all families of methods and big data technologies are also examined, including a review of the state-of-the-art. In addition, research challenges are discussed, with focus on developments on different big data framework, such as Hadoop, Spark and Flink and the encouragement in devoting substantial research efforts in some families of data preprocessing methods and applications on new big data learning paradigms.

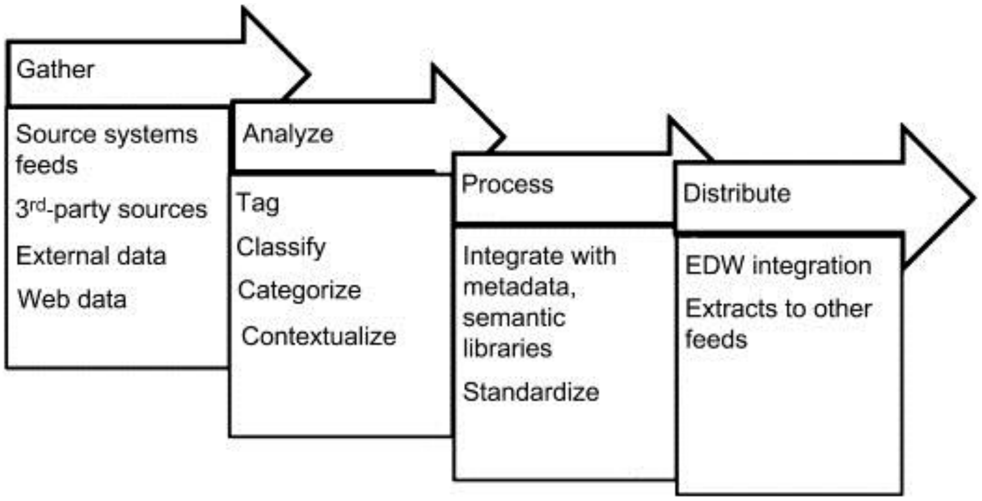

Figure 1.5 Different Stage of data

● Gather the data.

● Analyze the data.

● Process the data.

● Distribute the data.

While the stages are similar to traditional data processing the key differences are:

● Data is first analyzed and then processed.

● Data standardization occurs in the analyze stage, which forms the foundation for the distribute Stage where the data warehouse integration happens.

● there is not special emphasis on data quality except the use of metadata, master data, and semantic libraries to enhance and enrich the data.

● Data is prepared in the analyze stage for further processing and integration.

The stages and their activities are described in the following sections in detail, including the use of metadata, master data, and governance processes.

1.7.1 Gather Stage

Data is acquired from multiple sources including real-time systems, near-real-time systems, and batch-oriented applications. The data is collected and loaded to a storage environment like Hadoop or MySQL. Another option is to process the data through a knowledge discovery platform and store the output rather than the whole data set.

1.7.2 Analysis stage

The analysis stage is the data discovery stage for processing Big Data and preparing it for integration to the structured analytical platforms or the data warehouse. The analysis stage consists of tagging, classification, and categorization of data, which closely resembles the subject area creation data model definition stage in the data warehouse.

Tagging: A common practice that has been prevalent since 2003 on the Internet for data sharing. Tagging is the process of applying a term to an unstructured piece of information that will provide a metadata-like attribution to the data. Tagging creates a rich nonhierarchical data set that can be used to process the data downstream in the process stage.

Classify: Unstructured data comes from multiple sources and is stored in the gathering process. Classification helps to group data into subject-oriented data sets for ease of processing. For example, classifying all customer data in one group helps optimize the processing of unstructured customer data.

Categorize: the process of categorization is the external organization of data from a storage perspective where the data is physically grouped by both the classification and then the data type. Categorization will be useful in managing the life cycle of the data since the data is stored as a write-once model in the storage layer.

1.7.3 Process stage

Processing Big Data has several sub stages, and the data transformation at each sub stage is significant to produce the correct or incorrect output.

1.7.4 Context processing

Context processing relates to exploring the context of occurrence of data within the unstructured or Big Data environment. The relevancy of the context will help the processing of the appropriate metadata and master data set with the Big Data.

1.1 Advantages of Big Data

The vast majority of companies say that the benefits of big data are substantial. In the New Vantage Partners survey, 73.2 percent of executives said that they had seen measurable business results from their efforts. In addition, "Those executives who responded 'no' believe that it is too early to tell what impact these investments will have on their firm."

Enterprises report multiple advantages of big data, including the following:

- Better decision-making: In the New Vantage Partners survey, 36.2 percent of respondents said that better decision-making was the number one goal of their big data analytics efforts. In addition, 84.1 percent had started working toward that goal, and 59.0 percent had experienced some measurable success, for an overall success rate of 69.0 percent. Analytics can give business decision-makers the data-driven insights they need to help their companies compete and grow.

- Increased productivity: A separate survey from vendor Sync sort found that 59.9 percent of respondents were using big data tools like Hadoop and Spark to increase business user productivity. Modern big data tools are allowing analysts to analyze more data, more quickly, which increases their personal productivity. In addition, the insights gained from those analytics often allow organizations to increase productivity more broadly throughout the company.

- Reduce costs: Both the Sync sort and the New Vantage surveys found that big data analytics were helping companies decrease their expenses. Nearly six out of ten (59.4 percent) respondents told Sync sort big data tools had helped them increase operational efficiency and reduce costs, and about two thirds (66.7 percent) of respondents to the New Vantage survey said they had started using big data to decrease expenses. Interestingly, however, only 13.0 percent of respondents selected cost reduction as their primary goal for big data analytics, suggesting that for many this is merely a very welcome side benefit.

- Improved customer service: Among respondents to the New Vantage survey, improving customer service was the second most common primary goal for big data analytics projects, and 53.4 percent of companies had experienced some success in this regard. Social media, customer relationship management (CRM) systems and other points of customer contact give today's enterprises a wealth of information about their customers, and it is only natural that they would use this data to better serve those customers.

- Fraud detection: Another common use for big data analytics — particularly in the financial services industry — is fraud detection. One of the big advantages of big data analytics systems that rely on machine learning is that they are excellent at detecting patterns and anomalies. These abilities can give banks and credit card companies the ability to spot stolen credit cards or fraudulent purchases, often before the cardholder even knows that something is wrong.

- Increased revenue: When organizations use big data to improve their decision-making and improve their customer service, increased revenue is often the natural result. In the Sync sort survey, more than half of respondents (54.7 percent) said they were using big data tools to increase revenue and accelerate growth based on better insights.

- Increased agility: Again, from the Sync sort report, 41.7 percent of respondents said that one of the benefits of big data was the ability to increase business/IT agility. Many organizations are using their big data to better align their IT and business efforts, and they are using their analytics to support faster and more frequent changes to their business strategies and tactics.

- Greater innovation: Innovation is another common benefit of big data, and the New Vantage survey found that 11.6 percent of executives are investing in analytics primarily as a means to innovate and disrupt their markets. They reason that if they can glean insights that their competitors don't have, they may be able to get out ahead of the rest of the market with new products and services.

- Faster speed to market: Along those same lines, executives also told New Vantage that they were using big data to achieve faster time-to-market. Only 8.8 percent said that this was their number one goal for big data, but 53.6 percent have started working toward that goal, and of those, 54.1 percent had achieved some success. This advantage of big data is also likely to result in additional benefits, such as faster growth and higher revenue.

1.2 Disadvantages of Big Data

On the other side of the equation, many companies have also reported significant challenges when implementing big data analytics initiatives. Reported disadvantages of big data include the following:

- **Need for talent:** Data scientists and big data experts are among the most highly coveted and highly paid workers in the IT field. The At Scale survey found that the lack of a big data skill set has been the number one big data challenge for the past three years. And in the Sync sort survey, respondents ranked skills and staff as the second biggest challenge when creating a data lake. Hiring or training staff can increase costs considerably, and the process of acquiring big data skills can take considerable time.

- **Data quality:** In the Sync sort survey, the number one disadvantage to working with big data was the need to address data quality issues. Before they can use big data for analytics efforts, data scientists and analysts need to ensure that the information they are using is accurate, relevant and in the proper format for analysis. That slows the reporting process considerably, but if enterprises don't address data quality issues, they may find that the insights generated by their analytics are worthless or even harmful if acted upon.

- **Need for cultural change:** Many of the organizations that are utilizing big data analytics don't just want to get a little bit better at reporting, they want to use analytics to create a data-driven culture throughout the company. In fact, in the New Vantage survey, a full 98.6 percent of executives said that their firms were in the process of creating this new type of corporate culture. However, changing culture is a tall order. So far, only 32.4 percent were reporting success on this front.

- **Compliance:** Another thorny issue for big analytics efforts is complying with government regulations. Much of the information included in companies' big data stores is sensitive or personal, and that means the firm may need to ensure that they are meeting industry standards or government requirements when handling and storing the data. In the Sync sort survey, data governance, including compliance, was the third most significant barrier to working with big data. In fact, when respondents were asked to rank big data challenges on a scale from (most significant) to 5 (least significant), this disadvantage of big data got more 1s than another other challenge.

- **Cyber security risks:** Storing big data, particularly sensitive data, can make companies a more attractive target for cyberattackers. In the At Scale survey, respondents have consistently listed security as one of the top challenges of big data, and in the New Vantage report, executives ranked cyber security breaches as the single greatest data threat their companies face.

- **Rapid change:** Another potential drawback to big data analytics is that the technology is changing rapidly. Organizations face the very real possibility that they will invest in a particular technology only to have something much better come along a few months later. Sync sort respondents ranked this disadvantage of big data fourth among all the potential challenges they faced.

- **Hardware needs:** Another significant issue for organizations is the IT infrastructure necessary to support big data analytics initiatives. Storage space to house the data, networking bandwidth to transfer it to and from analytics systems, and compute resources to perform those analytics are all expensive to purchase and maintain. Some organizations can offset this problem by using cloud-based analytics, but that usually doesn't eliminate the infrastructure problems entirely.

- **Costs:** Many of today's big data tools rely on open source technology, which dramatically reduces software costs, but enterprises still face significant expenses related to staffing, hardware, maintenance and related services. It's not uncommon for big data analytics initiatives to run significantly over budget and to take more time to deploy than IT managers had originally anticipated.

- **Difficulty integrating legacy systems:** Most enterprises that have been around for very many years have siloed data in a variety of different applications and systems throughout their environments. Integrating all those disparate data sources and moving data where it needs to be also adds to the time and expense of working with big data.

Features	**Big Data**	**Hadoop**
Definations	Big data refers to the large volume of data both structured and unstructured.	Hadoop is framework to process large amount of data
Significance	Big data has no significant until it process.	It is tools that make a big data very meaningful by processing the data.
Storage	It is very difficult to store big data because its comes in both structured and unstructured data.	Apache hadoop HDFS is capable for storing the data
Accessibility	When it comes to accessing big data is very difficult.	Hadoop framework lets you access and process the data

Table 1.1 Difference between big data and hadoop

Chapter 2

Literature Review

2.1 Big Data Hadoop

Hadoop is a distributed system infrastructure researched and developed by the Apache Foundation. The users can develop the distributed applications although they do not know the lower distributed layer so that the users can make full use of the power of the cluster to perform the high-speed computing and storage. The core technologies of Hadoop are the Hadoop Distributed File System (HDFS) and the Map Reduce. HDFS provides the huge storage ability while Map Reduce provides the computing ability of the Big Data. Since HDFS and Map Reduce have become open source, their low cost but high processing performance helped them to be adopted by many enterprises and organizations. With the popularity of the Hadoop technologies, there are more tools and technologies which are developed on the basis of the Hadoop framework.

2.1.1 Relevant Hadoop Tools

Hadoop has developed a collection of many projects. Although the core technologies are HDFS and Map Reduce, Chukwa, Hive, Hbase, Pig, Zookeeper and so on are also indispensable. They provide the complementary services and higher level service on the core layer.

- Map Reduce is a programming model for parallel computing on the large-scale data sets. The concepts of Mapping and Reducing are referenced to the functional programming languages. The programmers who are not familiar with distributed parallel programming can also run their programs on the distributed system.
- HDFS is a distributed file system. Because is high fault-tolerant, it can be applied on the low-cost hardware. By providing the high throughput access to the data of the applications, it is suitable for applications which contain the huge data sets.
- Chukwa is an open source data collection system which is used for data monitoring and analysis. Chukwa is built on the HDFS and Map Reduce framework. Chukwa stores the data by HDFS and relies on the Map Reduce to manage the data. Chukwa is a flexible and powerful tool to display, monitoring, and analyze the data.
- Hive was originally designed by Facebook. It is a data warehouse based on the Hadoop which provides data sets searching, special query, and analysis. Hive is a structured data mechanism which supports the SQL query languages like RDBMS to

- Help those users who are familiar with SQL queries. This kind of query language is called Hive QL. In fact, the traditional Map Reduce programmers can query data on the

Mapper or Reducer by using Hive QL. The Hive compiler will compile the Hive QL into a Map Reduce task.

- Hbase is a distributed and open source database. As mentioned in Chapter2, the concepts of Hadoop originally came from Google; therefore, Hbase shares the same data model. Because the forms of Hbase are loose and the users can define various columns. Hbase is usually used for Big Data.
- Pig is a platform which was designed for the analysis and evaluation of the Big Data. The significant advantage of its structure is that it can afford the highly parallel test which is based on the parallel computing. Currently, the bottom layer of the Pig is composed of a complier. When the complier is running, it will generate some program sequences of Map Reduce.
- Zookeeper is an open source coordination service for the distributed applications. It is used mainly to provide users' synchronization, configuration management, grouping, and naming services. It can reduce the coordination tasks for the distributed applications.

2.2 Hadoop Distributed File System (HDFS)

HDFS is the master/slave structure. The Name node is the master node, while the Data node is the slave node. Documents are stored as data blocks in the Data node. The default size of a data block is 64M and it cannot be changed. If the files are less than a block data size, HDFS will not take up the whole block storage space. The Name node and the Data node normally run as Java programs in the Linux operating system. The Name node which is the manager of the HDFS is responsible for the management of the namespace in the file system. It will put all the folders and files metadata into a file system tree which maintains all the metadata of the files directories. At the same time, Name node also saves the corresponding relations between each file and the location of the data block. Data node is the place to store the real data in the system. However, all the data is not stored on the hard drives but will be collected when the system starts to find the resource data server of the required documents. The Secondary Name node is a backup node for the Name node. If there is only one Name node in the Hadoop cluster environment, the Name node will obviously become the weakest point of the process in the HDFS. Once the failure of the Name node occurs, it will affect the whole operation of the system. This is the reason why Hadoop designed the Secondary Name node as the alternative backup. The Secondary Name node usually runs on a separate physical computer and keeps communication at certain time interval to keep the snapshot of the file system metadata with the Name node so that it can recovery the data immediately in case some error happens. The Data node is the place where the real data is saved and handles most of the fault-tolerant mechanism. The files in HDFS are usually divided into multiple data blocks stored in the form of redundancy backup in the Data node. The Data node reports the data storage lists to the Name node regularly so that the user can obtain the data by directly access to the Data Node. The Client is the HDFS user. It can read and write the data though calling the API provided by HDFS. While in the read and write process, the client first needs to obtain the metadata a information from the Name node, and then the client can perform the corresponding read and write operations.

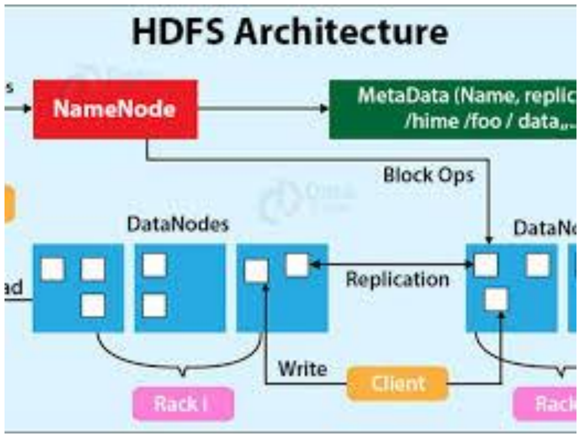

Figure 2.1 HDFS Architecture

2.2.1 Data Reading Process in HDFS

The data reading process in HDFS is not difficult. It is similar to the programming logic which has created the object, i.e., calling the method and performing the execution. The following section will introduce the reading processing of the HDFS. Figure2.1 HDFS reading process.

There are six steps when the HDFS has the reading process:

- The client will generate a DistributedFileSystem object of the HDFS class library and uses the open () interface to open a file.
- DistributedFileSystem sends the reading request to the Name node by using the Remote Procedure Call Protocol to obtain the location address of the data block. After the calculating and sorting the distance between the client and the Data node, the Name node will return the location information of the data block to the Distributed File System.
- After, the DistributedFileSystem has already received the distances and address of the data block; it will generate a FS Data Input Stream object instance to the client. At the same time, the FSDataInputStream also encapsulates a DFS Input Stream object which is responsible for saving the storing data blocks and the address of the Data node.
- After receiving the calling method, the encapsulated DFS Input Stream of FSDataInputStream will choose the nearest Data node to read and return the data to the client.

- When all the data has been read successfully, the DFSInputStream will be in charge of closing the link between the client and the Data node.

While the DFSInputStream is reading the data from the Data node, it is hard to avoid the failure that may be caused by network disconnection or node errors. When this happens, DFSInputStream will give up the failure Data node and select the nearest Data node. In the later reading process, the disfunctioning Data node will not be adopted anymore. It is observed that HDFS separates the index and data reading to the Name node and Data node. The Name node is in charge of the light file index functions while the heavy data reading is accomplished by several distributed Data nodes. This kind of platform can be easily be adapted to the multiple user access and huge data reading.

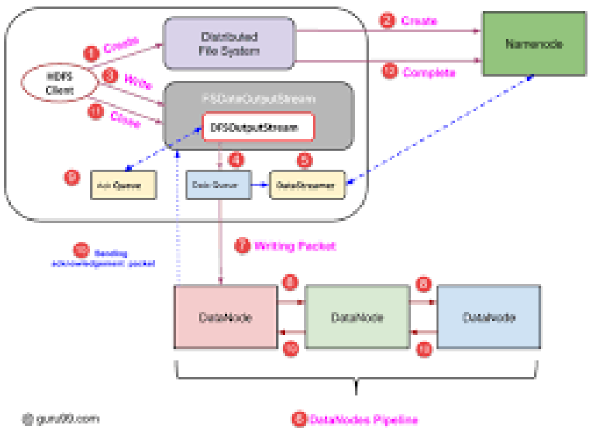

Figure 2.2 Data Reading Process

2.2.2 Data Writing Process in HDFS

- The data writing process in HDFS is the opposite process of the reading but the writing process is more complex. The following section will introduce the writing process in HDFS briefly
- The structure of HDFS reading process is similar to the writing process. There are the following seven steps:

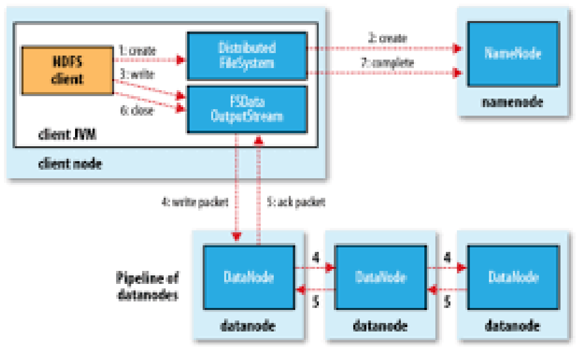

Fig. 2.3 Data Writing Process

- The client generates a DistributedFileSystem object of the HDFS class library and uses the create () interface to open a file.
- DistributedFileSystem sends the writing request to the Namenode by using the Remote Procedure Call Protocol (RPC). The Namenode will check if there is a duplicate file name in it. After that, the client with writing authority can create the corresponding records in the namespace. If an error occurrs, the Namenode will return the Exception to the client.
- After the DistributedFileSystem has received the successful return message from the Namenode, it will generate a FS Data Output Stream object to the client. In the FS Data Output Stream, there is an encapsulated DFS Output Stream object which is responsible for the writing process. The client calls the write () method and sends the data to the FSDataInputStream.
- For each data block, the Namenode will assign several Data nodes to store the data block. For instance, if one block needs to store in three Data nodes. Data Streamer will write the data block at the first Data node, then the first Data node will pass the data block to the second data node, and the second one passes to the third one. Finally, it will complete the writing data in the data node chain.
- After every data node has been written, data node will report to the Data Streamer. Step4 and Step5 will be repeated until all the data has been written successfully.
- When all the data has been written, the client will call the close () method of FSDataInputStream to close the writing operation.
- Finally, the Namenode will be informed by the DistributedFileSystem that all the written process has been completed.

In the process of data writing, if one data node makes error and causes writing failure, all the links between the Data Streamer and the data node will be closed. At the same time, the failure node will be deleted from the data node chain. The Namenode will notice the failure by the returned packages and will assign a new data node to continue the processing. As long as one data node is written successfully, the writing operation will regard the process as completed.

2.3 Authority management of HDFS

HDFS shares a similar authority system to POSIX. Each file or directory has on owner and a group. The authority permissions for the files or the directories are different to the owner, users in the same group, and other users. On the one hand, for the files, users are required the –r authority to read and the –w authority to write. On the other hand, for the directories, users need the –r authority to list the directory content and –w authority to create or delete. Unlike the POSIX system, there is no sticky, setuid or setgid of directories because there is no concept of executable files in HDFS.

2.4 Limitations of HDFS

HDFS as the open source implementation of GFS is an excellent distributed file system and has many advantages. HDFS was designed to run on the cheap commodity hardware not on expensive machines. This means that the probabilities of node failure are slightly high. To give a full consideration to the design of HDFS, we may find that HDFS has not only advantages but also limits for dealing with some specific problems. These limitations are mainly displayed in the following aspects:

- **High Access Latency**
  HDFS does not fit fort requests which should be applied in a short time. The HDFS was designed for the Big Data storage and it is mainly used for it high throughput abilities. This may cost the high latency instead. Because HDFS has only one single Master system, all the file requests need to be processed by the Master. When there is a huge number of requests, there is inevitably has the delay. Currently, there are some additional projects to address this limitation, such as using the Hbase uses the Upper Data Management project to manage the data.

- **Poor small files performance**
  HDFS needs to use the Namenode to manage the metadata of the file system to respond to the client and return the locations so that the limitation of a file size is determined by the Namenode. In general, each file, folder, and block need to occupy the 150 bytes' space. In other words, if there are one million files and each file occupies one block, it will take 300MB space. Based on the current technology, it is possible to manage millions of files. However, when the files extend to billions, the work pressures on the Namenode are heavier and the time of retrieving data is unacceptable.

2.5 MAP REDUCE

In 2004, Google published a thesis to introduce MapReduce. the greatest achievement of the MapReduce is that it can rewrite the Google's index file system. Until now, MapReduce is widely used in logs analysis, data sorting, and specific data searching. According to the Google's thesis, Hadoop implements the programming framework of the MapReduce and renders it an open source framework. MapReduce is the core technology of Hadoop. It provides a parallel computing model for the Big Data and supports a set of programming interfaces for the developer. Map Reduce is a standard functional programming model. This kind of model has been used in the early programming languages, such as Lisp. The core of the calculation model is that can pass the function as the parameter to another function. Through multiple concatenations of functions, the data processing can turn into a series of function execution. Map Reduce has two stage of processing. The first one is Map and the other one is Reduce.

The reason why the Map Reduce is popular is that it is very simple, easy to implement, and offers strong expansibility. Map Reduce is suitable for processing the Big Data because it can be processed by the multiple hosts at the same time to gain a faster speed.

2.5.1 MapReduce Architecture

The MapReduce operation architecture includes the following three basic components

- **Client:** Every job in the Client will be packaged into a JAR file which is stored in HDFS and the client submits the path to the Job Tracker
- **Job Tracker:** The Job Tracker is a master service which is responsible for coordinating all the jobs that are executed on the MapReduce. When the software is on, the Job Tracker is starting to receive the jobs and monitor them. The functions of MapReduce include designing the job execution plan, assigning the jobs to the Task Tracker, monitoring the tasks, and redistributing the failed tasks.
- **Task Tracker:** The Task Tracker is a slave service which runs on the multiple nodes. It is in charge of executing the jobs which are assigned by the Job Tracker. The Task Tracker receives the tasks through actively communicating with the Job Tracker.

2.5.3 MapReduce Procedure

The MapReduce procedure is complex and smart. This thesis will not discuss the MapReduce procedure in detail but will introduce it briefly based on author's own thoughts. Usually, MapReduce and HDFS are running in the same group of nodes. This means that the computing nodes and storage nodes are working together. This kind of design allows the framework to schedule the tasks quickly so that the whole cluster will be used efficiently. The process of MapReduce can be divided into the following six steps.

Job Submission When the user writes a program to create a new Job Client the Job Client will send the request to Job Tracker to obtain a new Job ID. Then, the Job Client will check if the input and output directories are correct. After this check, Job Client will store the related resources which contain the configuration files, the number of the input data fragmentations, and Mapper/Reducer JAR files to HDFS. In particular, the JAR files will be stored as multiple backups.

Job Initialization As the master node of the system, Job Tracker will receive several Job Client requests so that Job Tracker implements a queue mechanism to deal with these problems. All the requests will be in a queue that is managed by the job schedulerWhen the Job Tracker starts to initialize its job is to create a Job in Progress instance to represent the Job. The Job Tracker needs to retrieve the input data from HDFS and to decide on the number of the Map tasks. The Reduce tasks and the Task in Progress are determined by the parameters in the configuration files.

Task Allocation The task allocation mechanism in the MapReduce is to pull the whole process. Before the task allocation, the Task Tracker which is responsible for Map tasks and Reduce tasks has been already launched. The Task Tracker will send the heartbeat message to the Job Tracker to ask if there are any tasks that can be done any time. When the Job Tracker job queue is not empty, the Task Tracker will receive the tasks to do. Due to the lack of the Task Tracker computing capability, the tasks that can be done on the Task Tracker are also limited. Each Task Tracker has two fixed task slots which correspond to the Map tasks and Reduce tasks. During the tasks allocation, the Job Tracker will use the Map task slot first. Once the Map task slot is empty, it will be assigned to the next Map task. After the Map task slot is full, then the Reduce task slot revives the tasks to do.

Map Tasks Execution After the Map Task Tracker has received the Map tasks, there is a series of operations to finish the tasks. Firstly, the Map Task Tracker will create a Task In Progress object to schedule and monitor the tasks. Secondly, the Map Task Tracker will take out and copy the JAR files and the related parameter configuration files from HDFS to the local working directory. Finally, when all the preparations have been competed, the Task Tracker will create a new Task Runner to run the Map task. The Task Runner will launch a separate JVM and will start the Map Task inside to execute the map() function in case the abnormal Map Task affects the normal Task Tracker works. During the process, the Map Task will communicate with Task Tracker to report the task progress until all the tasks are completed. At that time, all the computing results will be stored in the local disk.

Reduce Tasks Execution When the part of the Map Tasks completed, the Job Tracker will follow a similar mechanism to allocate the tasks to the Reduce Task Tracker. Similar to the process of Map tasks, the Reduce Task Tracker will also execute the reduce() function in the separate JVM. At the same time, the Reduce Task will download the results data files from the Map Task Tracker. Until now, the real Reduce process has not started yet. Only when all the Map tasks have been completed, the Job Tracker will inform the Reduce Task Tracker to start to work. Similarly, the Reduce Task will communicate with the Task Tracker about the progress until the tasks are finished.

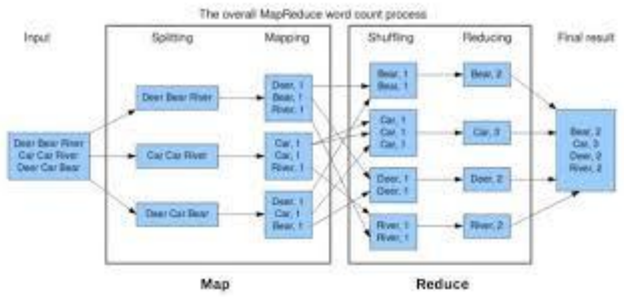

Fig 2.4 Map Reduce Processing

2.6 Limitations of MapReduce

Although MapReduce is popular all over the world, most people still have realized the limits of the MapReduce. There are following the four main limitations of the MapReduce:

- **The bottleneck of Job Tracker**
  The Job Tracker should be responsible for jobs allocation, management, and scheduling. In addition, it should also communicate with all the nodes to know the processing status. It is obvious that the Job Tracker which is unique in the Map Reduce, takes too many tasks. If the number of clusters and the submission jobs increase rapidly, it will cause network bandwidth consumption. As a result, the Job Tracker will reach bottleneck and this is the core risk of MapReduce. Because the jobs allocation information is too simple, the Task Tracker might assign a few tasks that need more sources or need a long execution time to the same node. In this situation, it will cause node failure or slow down the processing speed
- **Jobs Delay**
  Before the MapReduce starts to work, the Task Tracker will report its own resources and operation situation. According to the report, the Job Tracker will assign the jobs and then

the Task Tracker starts to run. As a consequence, the communication delay may make the Job Tracker to wait too long so that the jobs cannot be completed in time.

- **Inflexible Framework**
  Although the MapReduce currently allows the users to define its own functions for different processing stages, the MapReduce framework still limits the programming model and the resources allocation.

Hadoop Map Reduce	Yarn
Hadoop Map Reduce stores data on system	YARN is liable For asset the executives imply.
System based memory computing i.e. partial use of memory	its occupation will be executed by which framework get choose by YARN
It is for mass only	support interactive query
Map Reduce uses copying for fault tolerance	Introduce in hadoop 2.0
It is used for produce report help to find answer from historical queries	Yarn execution model is generic compare to map reduce
It is slower than yarn	It is 100 time faster than Map Reduce in Execution

Table 2.1 Difference between Hadoop and Yarn

2.7 Next generation of MapReduce: YARN

In order to solve above limitations, the designers have put forward the next generation of MapReduce: YARN. Given the limitations of MapReduce, the main purpose of YARN is to divide the tasks for the Job Tracker. In YARN, the resources are managed by the Resource Manager and the jobs are traced by the Application Master. The Task Tracker has become the Node Manager. Hence, the global Resource Manager and the local Node Manager compose the data computing framework. In YARN, the Resource Manager will be the resources distributor while the Application Master is responsible for the communication with the Resource Manager and cooperate with the Node Manager to complete the tasks.

2.7.1 YARN Architecture

Compared with the old MapReduce Architecture, it is easy to find out that YARN is more structured and simple. Then, the following section will introduce the YARN architecture.

- **Resource Manager**
  According to the different functions of the Resource Manager, it designers have divided it into two lower level components: The Scheduler and the Application Manager. On the one hand, the Scheduler assigns the resource to the different running applications based on the cluster size, queues, and resource constraints. The Scheduler is only responsible for the resources allocation but is not responsible for the monitoring the application implementation and task failure. On the other hand, the Application Manager is in charge of receiving jobs and redistributing the containers for the failure objects.

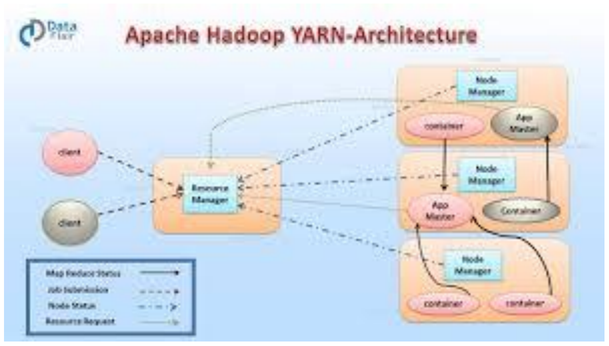

Fig 2.5 YARN Architecture

- **Node Manager**
  The Node Manager is the frame proxy for each node. It is responsible for launching the application container, monitoring the usage of the resource, and reporting all the information to the Scheduler.

- **Application Master**
  The Application Master is cooperating with the Node Manager to put tasks in the suitable containers to run the tasks and monitor the tasks. When the container has errors, the Application Master will apply for another resource from the Scheduler to continue the process.

- **Container**
  In YARN, the Container is the source unit which is the available node splitting the organization resources. Instead of the Map and Reduce source pools in MapReduce, the Application Master can apply for any numbers of the Container. Due to the same property Containers, all the Containers can be exchanged in the task execution to improve efficiency.

2.7.2 Advantages of YARN

Compared to the MapReduce, there are many advantages of the YARN framework. There are four main advantages of YARN compared to the Map Reduce.

- YARN greatly enhances the scalability and availability of the cluster by distributing the tasks to the Job Tracker. The Resource Manager and the Application Master greatly relieves the bottleneck of the Job Tracker and the safety problems in the Map Reduce.
- In YARN, the Application Master is a customized component. That means that the users can write their own program based on the programming model. This makes the YARN more flexible and suitable for wide use.
- YARN, on the one hand, supports the program to have a specific checkpoint. It can ensure that the Application Master can reboot immediately based on the status which was stored on HDFS. On the other hand, it uses the Zoo Keeper on the Resource Manager to implement the failover. When the Resource Manager revives errors, the backup Resource Manager will reboot quickly. These two measures improve the availability of YARN.

The cluster has the same Containers are the Reduce and Map pools in MapReduce. Once there is a request for resources, the Scheduler will assign the available resources in the cluster to the tasks and regard the resource type. It will increase the utilization of the cluster resources.

Chapter 3

Statement and Identification of problem

- Map reduce data locality is one of the key advantage of Hadoop Map Reduce the fact that the map code is executed on the same data node where the data resides. Interestingly many people found that this is not always the case in practice.
- Read Time degradation over time the Task Trackers and their effect on job performance, a slowdown of the Name Node or slowdowns of particular Data Nodes have a deteriorating effect on the whole cluster. Requests can easily be base lined, making the monitoring and degradation detection automatic.
- Some slow Nodes can bring down the cluster like every real world application; Cassandra Nodes can slow down due to many issues (hardware, compaction, GC, network, disk etc.). Cassandra is a clustered database where every row exists multiple times in the cluster and every write request is sent to all nodes that contain the row (even on consistency level one). It is no big deal if a single node fails because others have the same data all read and write requests can be fulfilled.
- Data distribution and placement to improve data locality, you need to first detect which of your jobs have a data locality problem or degrade over time. With APM solutions you can capture which tasks access which data nodes.
- Name Node and Data Node slowdowns the Task Trackers and their effect on job performance, a slowdown of the Name Node or slowdowns of particular Data Nodes have a deteriorating effect on the whole cluster. Requests can easily be baselined, making the monitoring and degradation detection automatic.
- Too many read round trips/Too much data read and typical performance problem with Cassandra reminds us of the SQL days and is typical for Cassandra beginners.It is a database design issue and leads to transactions that make too many requests per end-user transaction or read too much data.

3.1 Hadoop Issue Tracking

Hadoop tracks both bugs and enhancement requests using JIRA.

We welcome input, however, before filing a request, please make sure you do the following:

- Search the Apache JIRA database.
- Check the user's mailing lists, both by searching the archives and by asking questions. Common Hadoop Common issues are tracked in the HADOOP Jira instance.

HDFS HDFS issues are tracked in the HDFS Jira instance.

YARN YARN issues are tracked in the YARN Jira instance.

MapReduce Map Reduce issues are tracked in the MAPREDUCE Jira instance.

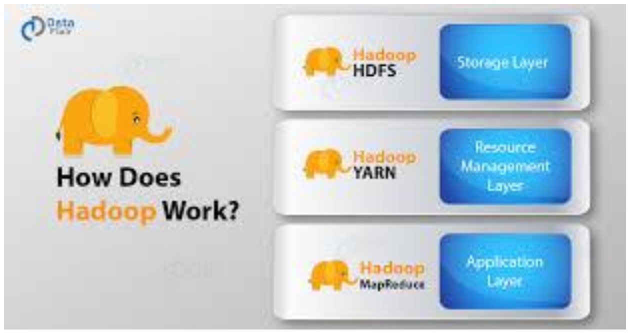

Figure 3.1 hadoop working

3.2 Hadoop Issues

3.2.1 Map Reduce data locality

Data locality is one of the key advantage of Hadoop Map/Reduce; the fact that the map code is executed on the same data node where the data resides. Interestingly many people found that this is not always the case in practice. Some of the reasons they stated were:

- Speculative execution
- Heterogeneous clusters
- Data distribution and placement
- Data Layout and Input Splitter

The challenge becomes more prevalent in larger clusters, meaning the more data nodes and data I have the less locality I get. Larger clusters tend not to be complete homogeneous, some nodes are newer and faster than others, bringing the data to compute ratio out of balance. Speculative execution will attempt to use compute power even though the data might not be local. The nodes that contain the data in question might be computing something else, leading to another node doing non-local processing. The root cause might also lie in the data layout/placement and the used Input Splitter. Whatever the reason non-local data processing puts a strain on the network which poses a problem to scalability. The network becomes the bottleneck. Additionally, the problem is hard to diagnose because it is not easy to see the data locality.

To improve data locality, you need to first detect which of your jobs have a data locality problem or degrade over time. With APM solutions you can capture which tasks access which data nodes. Solving the problem is more complex and can involve changing the data placement and data layout, using a different scheduler or simply changing the number of mapper and reducer slots for a job. Afterwards, you can verify whether a new execution of the same workload has a better data locality ratio.

3.2.2 Profiling Hadoop workloads

The next item confirmed our own views and is very interesting: many Hadoop workloads suffer from inefficiencies. It is important to note that this is not a critique on Hadoop but on the jobs that are run on it. However "profiling" jobs in larger Hadoop clusters is a major pain point. Black box monitoring is not enough and traditional profilers cannot deal with the distributed nature of a Hadoop cluster. Our solution to this problem was well received by a lot of experienced Hadoop developers. We also received a lot of interesting feedback on how to make our Hadoop job "profiling" even better.

3.2.3 Task Tracker performance and the impact on shuffle time

It is well known that shuffle is one of the main performance critical areas in any Hadoop job. Optimizing the amount of map intermediate data shuffle distribution and pure read/merge performance (number of threads, memory on the reducing side) are described in many Performance Tuning articles about Hadoop. Something that is less often talked about but is widely discussed by the long-term "Hadoopers" is the problem of a slowdown of particular TaskTrackers.When particular compute nodes are under high pressure, have degrading hardware, or run into cascading effects, the local Task Tracker can be negatively impacted. To put it in more simple terms, in larger systems some nodes will degrade in performance.

The result is that the Task Tracker nodes cannot deliver the shuffle data to the reducers as fast as they should or may react with errors while doing so. This has a negative impact on virtually all reducers and because shuffle is a choke point the entire job time can and will increase. While small clusters allow us to monitor the performance of the handful of running Task Trackers, real world clusters make that infeasible. Monitoring with Ganglia based on averages effectively hides which jobs trigger this, which are impacted and which Task Trackers are responsible and why.

The solution to this is a base lining approach, coupled with a Pure Path Pure Stack model. Base lining of Task Tracker requests solves the averaging and monitoring problem and will tell us immediately if we experience a degradation of Task Tracker map Output performance. By always knowing which Task Trackers slow down, we can correlate the underlying JVM host health and we are able to identify if that slowdown is due to infrastructure or Hadoop configuration issues or tied to a specific operating system version that we recently introduced.

Finally, by tracing all jobs, task attempts as well as all map Output requests from their respective task attempts and jobs we know which jobs may trigger a Task Tracker slowdown and which jobs suffer from it.

3.2.3 Name Node and Data Node slowdowns

Similar to the Task Trackers and their effect on job performance, a slowdown of the Name Node or slowdowns of particular Data Nodes have a deteriorating effect on the whole cluster. Requests can easily be base lined, making the monitoring and degradation detection automatic. Similarly, we can see which jobs and clients are impacted by the slowdown and the reason for the slowdown, be it infrastructure issues, high utilization or errors in the services.

`3.2.4 Some Slow Nodes Can Bring Down the Cluster

Like every real world application, Cassandra Nodes can slow down due to many issues (hardware, compaction, GC, network, disk etc.). It is a clustered database where every row exists multiple times in the cluster and every write request is sent to all nodes that contain the row (even on consistency level one). It is no big deal if a single node fails because others have the same data; all read and write requests can be fulfilled. In theory a super slow node should not be a problem unless we explicitly request data with consistency level "ALL," because Cassandra would return when the required amount of nodes responded. However internally every node has a coordinator queue that will wait for all requests to finish, even if it would respond to the client before that has happened. That queue can fill up due to one super slow node and would effectively render a single node unable to respond to any requests. This can quickly lead to a complete cluster not responding to any requests. The solution to this is twofold. If you can, use a token-aware client like Astana. By talking directly to the nodes containing the data item, this client effectively bypasses the coordinator problem. In addition you should baseline the response time of Cassandra requests on the server nodes and alert yourself if a node slows down. Funnily enough bringing down the node would solve the problem temporarily because Cassandra will deal with that issue nearly instantaneously.

3.2.5 Too many read round trips

Another typical performance problem with Cassandra reminds us of the SQL days and is typical for Cassandra beginners. It is a database design issue and leads to transactions that make too many requests per end-user transaction or read too much data. This is not a problem for Cassandra itself, but the simple fact of making many requests or reading more data slows down the actual transaction. While this issue can be easily monitored and discovered with an APM solution, the fix is not as trivial as in most cases it requires a change of code and the data model.

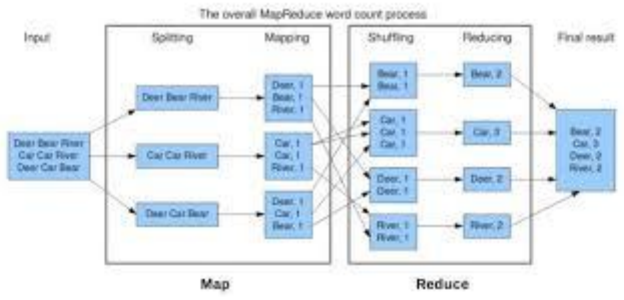

Figure 3.2 Map reduce Processing

3.3 Architectural security issues in hadoop

Hadoop, as we recognize, is an open-source venture that includes various modules, which are separately developed over time to add different types of functionalities to its core capabilities. Security was a late addition, and thus, Hadoop lacks a consistent security model.By default, Hadoop assumes a trusted environment. Hadoop has focused on improving its efficiency. Researchers are gradually paying attention to Hadoop security concerns and building security modules for it. However, currently, there is no existing evaluation for these Hadoop security modules. Due to huge volume, rapid growth, and diversity of data, these are unstoppable and existing security solutions are not adequate, which were not designed and build with Big Data in consideration. The Hadoop eco-system is a mixture of different applications including Pig, Hive, Flume, Oozie, HBase, Spark, and Strom. Each of these applications requires hardening to add security capabilities to a Big Data environment and functions to be scaled with the data. Bolt-on security doesn't scale well and easy. The security tools vendor have customizable offerings and applying a one point entry (gateway/perimeter) so that commands and data are entered into the cluster from single entry.

Hadoop is distinguished by its fundamentally different deployment model, which exhibits highly distributed, redundant, and elastic data repositories. However, the architecture of distributed computing present a unique set of following vulnerabilities and security threats for data center managers and security professionals.

- Distributed computing and fragmented data
- Node-to-Node communication and access to data
- Multiple interfaces

3.4 Security Threats and Possible Attacks

A threat is a potential danger to an information system. A threat is that someone, or something, will identify a specific vulnerability and use it against a company or individual to attack the system. The basic principle of security is CIA, which refers to confidentiality, integrity, and availability. Confidentiality is only possible if only an authenticated user can assess the system; it is also important to determine who should have access to the system and what resources a user can access. Hadoop security is divided into two levels. With respect to the time at level-1, the Hadoop system starts with a minimum security computing in only a trusted environment. The Hadoop system adds Kerberos for authentication as a perimeter security but not inside the cluster in level-2. Different security projects such as Apache Sentry, Apache KNOX, Apache Ranger, and Rhino are included in Hadoop at security phase Level-3. Kerberos also integrates with AD and LDAP. Each security product provides a discrete security solution due to the different purposes and functionality of each project. However, the Hadoop system is still required to move to security level-4 as a concrete solution for centralized security in one project.
How to investigate data leakage attacks in Hadoop is important but a long-neglected issue as per McAfee's report, which states that different threats to data have increased recently. The following section explains security threats that can hamper the operation of MapReduce and all framework components in the absence of a protected MapReduce environment.

- Authentication and authorization: Identity and authentication are central to any security effort. Without it we cannot determine who should have access to data and who should not. This is done at user level by full integration of Active directory (AD) and Lightweight Directory Access Protocol (LDAP) with Kerberos, and at service level with Role-Based Access Control (RBAC) at the node level.

- Administrative data access should be ACL's, File Permission and Segregation of administrative roles of OS administrators and Hadoop administrators.

- Hadoop stacks have many different components which come with default setting and no security. Configuration and patch management is required when running different configurations and patch levels at one time.

- There are no built-in monitoring tools to detect misuse or block malicious queries. Logs should configure to capture both the correct event types and sufficient information to determine user actions. Audit and Monitoring tools are important when volume and velocity of data are high.

- Applications are built on web service models, especially in social networks like Facebook, Yahoo and Snap Chat. Hadoop Web Applications may be vulnerable to well-known attacks due to insecure API's. Big Data cluster APIs need to be protected from usual web service attacks command injection, buffer overflow attacks, and all the other.

Chapter 4

Solution and Formulation of the Problem

4.1Potential Solutions for Data Volume Challenge

4.1.1 Hadoop

Tools like Hadoop are great for managing massive volumes of structured, semi-structured and unstructured data. Being a new technology and many professionals are unfamiliar with Hadoop. To use this technology, lots of resources are required to learn and this eventually diverts the attention from solving the main problem towards learning Hadoop.

4.1.2 Visualization

Another way to perform analyses and report but sometimes granularity of data increases the problem of accessing the detail level needed.

4.1.3 Robust Hardware

It is also a good way to handle volume problems. It enables increased memory and powerful parallel processing to chew high volumes of data swiftly.

4.1.4 Grid Computing

Grid computing is represented by a number of servers that are interconnected by a high speed network; each of the servers plays one or many roles. The two main benefits of Grid computing are the high storage capability and the processing power, which translates to the data and computational grids.

4.1.4 Spark

Platforms like Spark use model plus in-memory computing to create huge performance gains for high volume and diversified data. All these approaches allow firms and organizations to explore huge data volumes and get business insights from it. There are two possible ways to deal with volume problem. We can either shrink the data or invest in good infrastructure to solve the problem of data volume and based on our cost budget and requirements we can select technologies and methods described above. If we have resources with expertise in Hadoop.

4.2. Potential Solutions for Data Variety Problem

4.2.1. OLAP Tools (On-line Analytical Processing Tools)

Data processing can be done using OLAP tools and it establishes connection between information and It eventually assembles data into a logical way in order to access it easily and OLAP tools specialists can achieve high speed and low lagging time for processing high volume data, OLAP tools process all the data provided to them no matter they are relevant.

4.2.2. Apache Hadoop

It is open source software and its main purpose is to manage huge amounts of data in a very short span of time with great ease. The functionality of Hadoop is to divide data among Multiple systems infrastructure for processing it.

4.2.3. SAP HANA

SAP HANA is an in-memory data platform that is deployable as an on-premise appliance, or
In the cloud. It is a revolutionary platform that's best suited for performing real-time analytics, and developing and deploying real-time applications. New DB and indexing architectures make sense of disparate data sources swiftly.

4.2.4. Redundant physical infrastructure

The flawless IT infrastructure greatly enhances Big Data implementation after determining
And setting all the requirements against each of the following criteria:

- Performance
- Availability
- Scalability

We can resolve the variety of problems using ETL tools, visualization tools, and OLAP tools And by having the robust infrastructure. It is hard to say just one among them could solely Resolve the problem or more than one is needed or there could be some algorithm which Could synchronize the data varieties in a uniform format, it depends on particular case or Problem. We can’t have a generalized solution for all.

4.3 Potential Solutions for Velocity Problem

4.3.1. Flash memory

It is needed for caching data, especially in dynamic solutions that can parse that data as either Hot (highly accessed data) or cold (rarely accessed data).

4.3.2. Transactional databases

They are equipped with real-time analytics, faster response to decision making.
“A transactional database is a database management system that has the capability to roll

4.3.3. Cloud using hybrid model

Expanding private cloud using a hybrid model allows erupting for the additional Computational power needed for data analysis and to select hardware, software, and business
Process changes to handle high-pace data needs.

4.3.4. Sampling data

Sampling is a process (“Statistical analysis techniques are various statistics approaches used to select, manipulate and examine an elective subset of data points in order torecognizePatterns and inclinations in the massive data set being inspected.”) This makes data And can help deal well with issues like volume and velocity.

Hybrid SAAS/PAAS/LAAS systems along with cloud computation and storage can greatly
Solve the velocity problem but at the same time, we should take care of security and privacy
Of data.

4.4. Potential Solutions for the Quality Challenge

4.4.1. Data visualization

If the data quality is concerned, Visualization acts as an effective way because visualization
Of data lets we see where outliers and non-required data lie. For quality issue, firms should
Have a data control, surveillance or information management process active to ensure that the
Data is clean.

4.4.2 Big data algorithms

- Data quality and relevance is not a new concern, it is from the time we start dealing with dataAnd the storage of every piece of data a firm produces creates a problem. It is too expensiveto have dirty data and it costs companies in the United States hundreds of billions of dollars
- Every year. In addition to being perfect for maintaining, managing and cleaning data, big data Algorithms can be an easy step to clean the data. There are many algorithms and models or We can make our own algorithms to act on data to clean them and make them generous.

4.5. Potential Solutions for Privacy and Security Challenge

4.5.1. Examine your cloud providers

Storing big data in the cloud is a good way of storage along with this we need to take care of its protection mechanisms. We should make sure that our cloud provider must have frequent
Security audits and has disclaimer that include paying penalties in case adequate security standards have not met.

4.5.2 Must have an adequate access control policy

Create policies in such a way that allow access to authorized users only.

4.5.3 Protect the data

All stages of data should be adequately protected from the raw data, cleaned data to the final outcome from analytics. There should be encryption to ensure no sensitive data is leaked. The main solution to ensure that data remains protected is the adequate use of encryption. For example, Attribute-Based Encryption (type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes.) provides gauzy access control of encrypted data.

4.5.4 Protect communications

Data to data communication should also be adequately protected to ensure its confidentiality and integrity.

4.5.5 Acquire real-time security monitoring

Surveillance or monitoring should be there for access to the data. Threat inspection and Intelligence should be used to prevent unauthorized access to the data.

4.5.6 Anonym zings the data

There should be assurance that all sensitive data is removed from the set of records collected Before initiating data processing and analyses.

4.5.7 Real-time security monitoring

Organizations should monitor access to ensure that there is no unauthorized access. Threat Intelligence should be in place to ensure that attacks are detected and that the organizations can react to threats quickly.

4.5.8. Using authentication methods

Authentication is the process of verifying user or system identity before accessing the system.

4.5.9. Implementing access controls

Authorization is a method of indicating access control advantages for user or system to enhance security.

4.5.10. Use key management

File layer encryption is insufficient if an attacker or a hacker can access encryption keys. In most highly risky and insecure as keys can be retrieved easily by the platform administrator or an attacker so, Using key management service for keys distribution instead of just keeping one key and allotting different certificates and managing different keys for each group, application, interface and user is the most safe way to approach towards key thefts. of the cases administrators store keys on local disk drives for quick and easy access, but it’s

4.5.11. Logging

To detect attacks, failures, or unusual behavior, the activity log is very important and big data is the best fit for collecting and managing event data. Log files give a look up when something fails, or get hacked. So to meet the security and privacy needs, we need to audit the entire system on a periodic basis.

4.5.12. Use secure communication

Implement secure communication between nodes, interfaces, processes and applications.SSL/TLS implementation that protects all network communications rather than just a subset. Privacy of data is a huge concern everywhere. Private data can be misused and inappropriateness of use of personal data can be of any limit, particularly through linking of data from multiple reservoirs of data. So, unauthorized and unauthenticated use of data should be highly protected.

There are two common approaches to protect security and privacy of big data:

- First is to restrict access to the data by adding certification or access control to the data entries. We need to develop and design secured certification or access control mechanisms, such that no crucial information can be accessible to anyone apart from authorized users.

- The other approach is to anonymize data fields such that sensitive information cannot be recognized to a person's record. For that, we need to inject randomness into the data to ensure that it satisfies all the privacy goals.

4.6 Potential Solutions for Scalability challenge

4.6.1. Cloud Computing

A solution in the cloud will scale in much easier and faster way as compared to an on premises solution. Keeping data secured in a cloud can solve half a problem because the cloud can be secured and cloud can be highly spaced that we can call it nearly unlimited.

4.7 Open Issues in Big Data

4.7.1 Finding relevant data

There is a need for new metrics, techniques, analysis for finding relevant data. E.g. selecting Representative (diversity).

4.7.2 Machine Learning

There is a need to better integrate HPC ("High performance computing refers to any Computational activity requiring more than a single computer to execute a task. Supercomputers and computer clusters are used to solve advanced computation problems.") and Data Analytics (Machine learning) .

4.7.3 Need of Data Management and Data Analytics

Both management and analytics require different software and hardware characteristics, therefore, we will need machines that do support parallel machine learning to do analytics and Management together.

4.7.4 Need for Efficient Parallel Algorithms

There is need of developing efficient parallel algorithms for purpose of streaming and multiscale adaptive algorithms which reduce time complexity.

4.7.5 Need of Secure Cloud

There is a need for solutions to protect and securely process data in the cloud.

4.7.6 Need for Better Adaptations to Classical Algorithms

There is a need for better adaptations to classical algorithms to tackle big data challenges without many efforts.

4.7.7 Need for Better Transfer of Data from One Location to Another

Shipping external hard disks – processing the data while it is being transferred.

4.7.8 Security and Privacy of the Data

We need to use advanced encryption algorithms.

4.8 Big Data Hadoop Tools

4.8.1. Hadoop and MapReduce

It is a programming model for processing massive datasets and works on the principle of divide and conquer. The divide and conquer method is instrumented in two footsteps that are Map step and Reduce Step. Master node and worker node are two kinds of nodes. Division of input into tinier sub-problems and then disseminate them to worker nodes in map step. Eventually, the outputs for the entire sub-problems are combined by the master node in reduce step. Moreover, Hadoop and MapReduce are helpful in fault-tolerant storage and high output data processing.

4.8.2. Apache Mahout

It provides scalable and commercial machine learning techniques for big data and smart data analysis. Clustering, classification, pattern mining, regression, dimensionality reduction, evolutionary algorithms, and batch based collaborative filtering are core algorithms of the mahout. Apache mahout is to provide a tool for attenuating big challenges.

4.8.3. Apache Spark

It is a framework that provides speed processing, and sophisticated analytics. It lets us write Applications in java, scale, or python. It supports MapReduce operations, SQL queries, streaming data, machine learning, and graphs data processing. It consists of three components which are driver program, cluster manager, and worker nodes. The driver program acts as the main application point of the application on the spark cluster. The cluster manager assigns the Resources and the worker nodes to do the data processing in the form of tasks. Deploying spark applications in an existing Hadoop clusters is main advantage. Resilient distributed datasets (RDD) is the main emphasis of Spark; RDD stores data in-memory and provide fault tolerance without replication. It supports iterative computation, enhances speed and does better resource utilization. It also supports streaming data, machine learning, and graph algorithms.

4.8.4. Dryad

It is a programming model for executing parallel and distributed programs for managing large Databases. It has a cluster of computing nodes. It is an infrastructure for running data-parallel Programs. Dryad uses various machines with different and several processors and performs Concurrent programming.

4.8.5. Storm

It a distributed and computation system for processing unbounded streaming data. It designed for real-time processing instead of batch processing. It is very handy and easy to operate and has great performance. It is a free and open source distributed real-time computation system.

4.8.6. Apache Drill

It is a distributed system for interactive analysis of big data. It supports many query languages, several data formats. It is specifically for exploiting nested data. It has an amazing capability to scale up to thousands of servers or even more and has the capability to process massive data (petabytes to zeta bytes) and trillions of records in few seconds. The drill has to use HDFS for storage and map reduce for batch analysis.

4.8.7. Jasper soft

It is open source software that is mainly for producing reports from database columns. It's Basically an analytical platform and provides data visualization for storage platforms, including Mango DB, Cassandra, and much more. There is hardly any tool or software that can analyze that without extraction, transformation, and loading (ETL) but Jasper soft does. It is super cool that It can create HTML reports and dashboards directly from big data store without ETL. It is an absolute reporting tool and reports generated by this tool can be shared anywhere.

4.8.8. Splunk

It is a real-time and smart platform created for making the best use of machine-generated big data. It merges the up-to-the-moment cloud technologies and big data very swiftly and easily. It provides a web interface to help users monitor their machine-generated data. Splunk generates outputs in the form of graphs, reports, and alerts; it is used for indexing all kinds of data including structured, semi-structured and unstructured. It is good at resolving problems for system and information technology infrastructures.

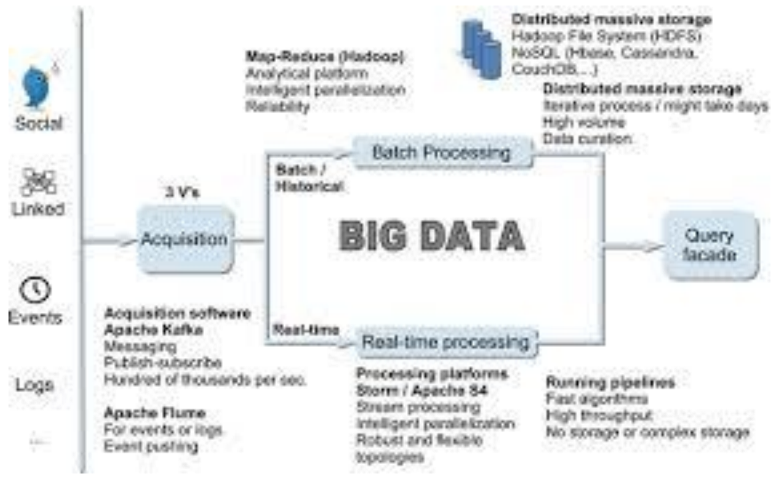

Fig 4.1 Big Data Acquition

4.9 Architectural security issues in hadoop

Hadoop, as we recognize, is an open-source venture that includes various modules, which are separately developed over time to add different types of functionalities to its core capabilities. Security was a late addition, and thus, Hadoop lacks a consistent security model. By default, Hadoop assumes a trusted environment. Hadoop has focused on improving its efficiency. Researchers are gradually paying attention to Hadoop security concerns and building security modules for it. However, currently, there is no existing evaluation for these Hadoop security modules.

Due to huge volume, rapid growth, and diversity of data, these are unstoppable and existing security solutions are not adequate, which were not designed and build with Big Data in consideration. The Hadoop eco-system is a mixture of different applications including Pig, Hive, Flume, Oozie, HBase, Spark, and Strom. Each of these applications require hardening to add security capabilities to a Big Data environment and functions to be scaled with the data. Bolt-on security doesn't scale well and easy. The security tools vendor have customizable offerings and applying a one point entry (gateway/perimeter) so that commands and data are entered into the cluster from single entry.

Hadoop is distinguished by its fundamentally different deployment model, which exhibits highly distributed, redundant, and elastic data repositories. However, the architecture of distributed computing present a unique set of following vulnerabilities and security threats for data center managers and security professionals.

- Distributed computing and fragmented data
- Node-to-Node communication and access to data
- Multiple interfaces

4.10 Security threats and possible attacks

A threat is a potential danger to an information system. A threat is that someone, or something, will identify a specific vulnerability and use it against a company or individual to attack the system. The basic principle of security is CIA, which refers to confidentiality, integrity, and availability. Confidentiality is only possible if only an authenticated user can assess the system; it is also important to determine who should have access to the system and what resources a user can access. Hadoop security is divided into two levels. With respect to the time at level-1, the Hadoop system starts with a minimum security computing in only a trusted environment. The Hadoop system adds Kerberos for authentication as a perimeter security but not inside the cluster in level-2. Different security projects such as Apache Sentry, Apache KNOX, Apache Ranger, and Rhino are included in Hadoop at security phase Level-3. Kerberos also integrates with AD and LDAP. Each security product provides a discrete security solution due to the different purposes and functionality of each project.

However, the Hadoop system is still required to move to security level-4 as a concrete solution for centralized security in one project. How to investigate data leakage attacks in Hadoop is important but a long-neglected issue as per McAfee's report, which states that different threats to data have increased recently. The following section explains security threats that can hamper the operation of MapReduce and all framework components in the absence of a protected MapReduce environment. A dispersed and replicated data processing of MapReduce can unlock an opportunity for a large range of attacks.

- Authentication and authorization: Identity and authentication are central to any security effort. Without it we cannot determine who should have access to data and who should not. This is done at user level by full integration of Active directory (AD) and Lightweight Directory Access Protocol (LDAP) with Kerberos, and at service level with Role-Based Access Control (RBAC) at the node level.
- Administrative data access should be ACL's, File Permission and Segregation of administrative roles of OS administrators and Hadoop administrators. Hadoop stacks have many different components which come with default setting and no security.

Configuration and patch management is required when running different configurations and patch levels at one time.

- There are no built-in monitoring tools to detect misuse or block malicious queries. Logs should configure to capture both the correct event types and sufficient information to determine user actions.

- Applications are built on web service models, especially in social networks like Facebook, Yahoo and Snap Chat. Hadoop Web Applications may be vulnerable to well-known attacks due to insecure API's. Big Data cluster APIs need to be protected from usual web service attacks command injection, buffer overflow attacks, and all the other.

4.11. Impersonation attacks

Impersonation attacks occur when an attacker tries to access resources by impersonating as an authorized identity. The attacker steals the credentials of an authorized user by using different methods and attacks Hadoop cluster resources, services, and jobs. Impersonation can be performed by replaying the tickets issued by the authentication server (Kerberos) for different purposes. Once the attacker gets access to the cluster, they can perform any type of data leak and slow down the processing of MapReduce.

4.12. Denial of Service(DoS)

A DoS attack occurs when the resources are unavailable to authorized users. As per the Verizon 2017 Data Breaches Report, attack incidents have been reported and out of those 5 breaches have been successful. DoS attacks are accomplished by flooding the target with traffic or cause a crash of data. In each instances, the DoS attack deprives legitimate users of the service or resource they expect. There are two general ways of DoS attacks: flooding services or crashing services. Flood attacks occur once the system receives an excessive amount of traffic for the server to buffer, inflicting it to abate and eventually stop. Well-known flood attacks embody distributed denial of service (DoS), buffer overflow, SYN flood, Ping to Death, and HTTP flood. In Hadoop, the Name Node and authentication server are vulnerable to DoS attacks. The Name Node has a master daemon that is responsible for scheduling and coordinating the execution of MapReduce applications on the data nodes. A DoS attack on the Name Node can halt all computations of MapReduce and read-write operations of HDFS.

4.13. Cross –Site scripting (XSS)

XSS is a common attack which injects a malicious code into a vulnerable web application. It differs from SQL injection because in that it does not directly target the application itself. But instead, the web application users are the ones at risk. XSS attacks can be broken down into two types: stored and reflected. Stored XSS, also known as persistent XSS, is more damaging than reflected XSS and occurs when a malicious script is injected directly into a vulnerable web application. Reflected XSS involves reflecting a malicious script of a web application onto a user's browser. The script is embedded into a link and is only activated once that link is clicked on. Hadoop web UI's are vulnerable to attacks, and recently, different database installations have been attacked.

4.14. Recent attacks

Hadoop is reportedly easily identifiable to hackers worldwide by simply sniffing to open instances, and around 5307 Hadoop clusters are exposed to the web with their security settings off, exploit them with receptive attack by hackers. The online search engine Shodan shows details about all servers and devices connected to the Internet. The merit of this is security recommendation, while the demerit is that it is used by hackers to see all details of any network, server location, and other settings. How to protect data attacks in Hadoop is an important issue. To counter these attacks and stop data theft, it is first important to thoroughly understand the vulnerabilities as well as threats, and then, it can be worked toward devising a strategy using defense-in-depth to secure data.

4.10 Security tools for hadoop cluster

4.10.1 Apache KNOX

Apache Knox Gateway is a single access point to a single or multiple Hadoop clusters based on the concept of the stateless reverse proxy framework. It also provides authentication to a group of authenticated users, authorization, auditing, and system monitoring. Knox hides data and details of Hadoop cluster installations, simplifies the amount of services that user have to be compelled to move with, and limits the numbers of access points with single entrance URL. Knox has a REST API -based perimeter security entrance system that authenticates user credentials against AD/LDAP and solely a successfully authenticated user is allowed access to the Hadoop cluster.

4.10.2 Apache Sentry

Apache Sentry is a granular, role-based authorization and a multi-tenant administration module for Hadoop. Sentry can manage access to data and metadata by enforcing an accurate level of privileges to authenticated users and applications in a Hadoop cluster. It is extraordinarily standard and might support authorization for a broad vary of data models in Hadoop. It is versatile and permits the definition of authorization rules to validate a user's or application's access requests for Hadoop resources. Sentry is meant to be a pluggable authorization engine for Hadoop elements like Apache Hive, Apache Solar, Impala, and HDFS.

4.10.3 Apache Ranger

Apache Ranger provides a centralized framework for securing Hadoop. Ranger is associate authorization system that allows/denies access to Hadoop cluster resources (HDFS files, Hive tables, etc.) supported pre-defined Ranger policies to authenticated users. When a user request comes to Ranger, it is assumed to have been already authenticated.

Apache Ranger uses Kerberos for authentication and Apache Knox for authorization: role -based access control (RBAC). Apache Knox also supports auditing of HDFS, Hive, and HBase, and Apache Ranger uses wire encryption for data protection.

4.10.4 Project Rhino

Project Rhino provides data protection to Hadoop stack with the single-sign-on (SSO) concept and supports encryption, a common authentication –authorization module key management. Rhino enhances cell-level encryption and fine-grained access control to HBase 0.98 and encryption to data-at-rest in Apache Hadoop. Data encryption in Hadoop requires both data-at-rest and data-in-transit; however, most Hadoop components provide encryption for data-in-transit only.

4.10.5 Kerberos

Kerberos is an authentication protocol developed by MIT. It is used in Hadoop to provide authentication to the user to access a Hadoop cluster. The Kerberos protocol uses secret-key cryptography to provide secure communications over a non -secure network. Kerberos is an SSO ticket-based system that relies on KDC. The Kerberos protocol generates three types of tickets for authentication: delegation token is a secret key between a user and Name Node for authentication, block access token is used to access a file from HDFS authenticated by Name Node and Data Node jointly to access a data block on the Data Node, and job token is generated by Job Tracker to authenticate tasks at Task Trackers. Kerberos KDC comprises of three components: an authentication server, a ticket -granting server (TGS), and a database.

Chapter 5

Implementation and Discussion

5.1 Big Data hadoop Implementation

5.1.1 Big Data: Definition and Types

Before talking about complicated projects, let’s start with a definition of big data. As we already told you, it is simple. Big Data is a very large and diverse set of information which is constantly growing. Big data can be both structured and unstructured. When structured, big data is organized in databases and data warehouses. Unstructured data can be defined as raw information collected by a company to understand the needs of its customers. It has no format or model to follow. Big data can be gathered from shared comments on websites and social networks, questionnaires, personal electronics, IoT and so one. There are a lot of potential sources of information.

5.1.2 Big Data hadoop Implementation Strategy

Big data can become one of your company’s most valuable resources. However, it won’t be able to play its role unless it is identified, gathered, managed and analyzed. To deal with this challenge, you need a reliable analytics strategy. Here are a few recommendations on developing such strategy:

5.1.3 Focus on customer-centric outcomes

Reaching success is impossible when your customers are unhappy. So when starting working on your strategy, prioritize customer-centric outcomes. These mean providing better services and increasing the customer retention rate.

5.1.3 Keep the entire company in mind

Your analytics strategy shouldn’t cover a single department of your company it should be suitable for the entire enterprise. A strategy must fit the vision of the company and its already existing strategy.

5.1.4 Start with the data which is already available

Before collecting the data, start with the information you already have. It may provide you only with short-term results, but this is still a wise decision. You will save some time and funds, and learn how to work with loads of data.

5.1.5 Invest in big data tools and skills for implementation

The number of analytics tools is constantly growing. Therefore, to create a strategy and implement a project, you will have to choose the most suitable ones. Don’t be afraid to invest money in them reliable tools are definitely worth the expenses. Besides, make sure that your data scientists know how to use the chosen tools, and invest in their education, if needed.

5.2 The Roadmap of the Analytical Big Data implementation Process

To implement your big data project, follow our roadmap. It is not ultimate, but it includes the most crucial steps you should take. Take a closer look.

5.2.1. Find a team and a sponsor

If you already have a dedicated team that can deal with the project, that’s great. If no, start with looking for experts. Besides, you may also need a sponsor. Big data projects take a lot of time and effort, which can go beyond your budget. So, calculate your potential expenses and decide if you need sponsorship or not.

5.2.2. Identify data sources

As we already told you, it is okay to start with already existing data. But you will still have to identify additional data sources you can use to collect the data. Identify, prioritize and evaluate them during this stage.

By the way, the data can be kept in so-called data lakes. A data lake is a repository for storing both structured and unstructured data. Unlike a data warehouse, a lake implies a flat architecture for storing the data. Besides, you can build and deploy data lakes using cloud infrastructure or on-premises infrastructure.

5.2.3. Connect data sources to your clients

After you are done with deciding on data sources, connect them to the needs of your clients. Let’s say, if you are leading the retail chain, you can collect the data with the help of digital coupons. A customer gets a coupon for a discount and is happy to visit your place. In turn, you also get what you want.

5.2.4. Incorporate new data hubs

Now, incorporate new data hubs one by one. Don’t hurry, this process should be gradual. In this way, you will have enough time to adjust your operations and understand how to use the data. If you try to do everything at once, your project can simply fail, so let things be incremental.

5.2.5. Connect the clients’ data to your company’s processes

Every data set you gather provides you with an opportunity to improve your services or products. Therefore, data-driven decisions should be present at the company at all levels: product development, pricing, marketing, etc.

5.2.6. Don’t forget about testing

Test, measure and learn that’s one of the most important things in the analytical process. When collecting another data set, test the related assumptions to understand how to move forward.
And one more tip here. To simplify working with the data and the project implementation, use big data visualization tools and techniques. They will provide you with a better understanding of large volumes of data, so the results will be more efficient.

5.3 Important Questions to Answer before Big Data implementation

Big Data always means loads of information, so when you realize the volume of the information gathered you will have to answer the following questions:

- Which data is relevant and suitable for the project?
- Which data doesn’t meet the relevance requirements?
- Is the data at rest? Just in case, data at rest is inactive data stored in any digital form.
- Is the data in motion? In turn, data in motion is data moving through the network.
- How can the data help you to achieve your goals? How can it serve you?
- Answering all these questions will help you to shape your project and maybe even save some funds.

A big data platform is a solution combining the capabilities of several utilities and tools for managing and analyzing the data. With a platform, you won’t have to use a lot of applications or tools it will work as a packaged solution. However, there is no universal platform to work with every single data set. You will have to develop a new one which will use the most effective tools for analyzing the data. Apart from this, your platform must fit your company's strengths.

5.4 Hadoop Implementation- Map Reduce

Map Reduce is a framework that is used for writing applications to process huge volumes of data on large clusters of commodity hardware in a reliable manner. This topic takes you through the operation of Map Reduce in Hadoop framework using Java.

5.4.1 Map Reduce Algorithm

Generally Map Reduce paradigm is based on sending map-reduce programs to computers where the actual data resides.

- During a Map Reduce job, Hadoop sends Map and Reduce tasks to appropriate servers in the cluster.
- The framework manages all the details of data-passing like issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on the nodes with data on local disks that reduces the network traffic.
- After completing a given task, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

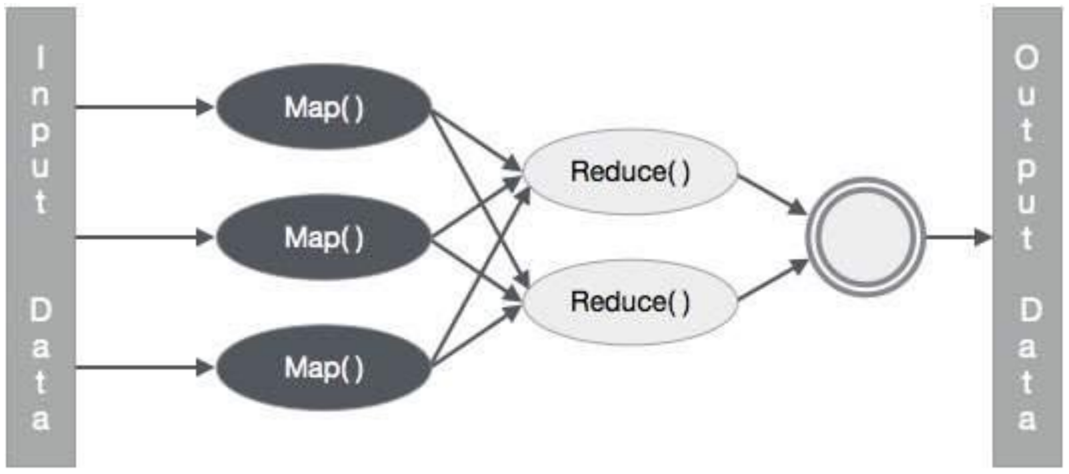

Figure 5.1 Map reduce Algorithm

5.4.2 Inputs and Outputs (Java Perspective)

The Map Reduce framework operates on key-value pairs, that is, the framework views the input to the job as a set of key-value pairs and produces a set of key-value pair as the output of the job, conceivably of different types.

The key and value classes have to be serializable by the framework and hence, it is required to implement the Writable interface. Additionally, the key classes have to implement the Writable Comparable interface to facilitate sorting by the framework.

Both the input and output format of a Map Reduce job are in the form of key-value pairs −

(Input) <k1, v1> -> map -> <k2, v2>-> reduce -> <k3, v3> (Output).

	Input	Output
Map	<k1, v1>	<k2, v2>
Reduce	<k2, list(v2)>	list (<k3, v3>

Table 5.1 Input and Output Map reduce

5.5 Map Reduce Implementation

The following table shows the data regarding the electrical consumption of an organization. The table includes the monthly electrical consumption and the annual average for five consecutive years.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Avg
1979	23	23	2	43	24	25	26	26	26	26	25	26	25
1980	26	27	28	28	28	30	31	31	31	30	30	30	29
1981	31	32	32	32	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	39	39	45

Table 5.2 Map reduce Implementation

We need to write applications to process the input data in the given table to find the year of maximum usage, the year of minimum usage, and so on. This task is easy for programmers with finite amount of records, as they will simply write the logic to produce the required output, and pass the data to the written application.

Let us now raise the scale of the input data. Assume we have to analyze the electrical consumption of all the large-scale industries of a particular state. When we write applications to process such bulk data,

- They will take a lot of time to execute.
- There will be heavy network traffic when we move data from the source to the network server.

To solve these problems, we have the Map Reduce framework.

5.5.1 Input Data

The above data is saved as **sample.txt** and given as input. The input file looks as shown below.

1979	23	23	2	48	24	25	26	26	26	26	25	26	25
1980	26	27	28	23	28	30	31	31	31	30	30	30	29

1981	31	32	32	28	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	40	39	45

Table 5.3 Sample Input Text

5.6 Map Reduce - User Interfaces

This section provides a reasonable amount of detail on every user-facing aspect of the Map Reduce framework. This should help users implement, configure and tune their jobs in a fine-grained manner. However, please note that the javadoc for each class/interface remains the most comprehensive documentation available; this is only meant to be a tutorial.

Let us first take the Mapper and Reducer interfaces. Applications typically implement them to provide the map and reduce methods. We will then discuss other core interfaces including Job, Partitioned, Input Format, Output Format, and others. Finally, we will wrap up by discussing some useful features of the framework such as the Distributed Cache, Isolation Runner etc.

5.6.1 Mapper

- Mapper maps input key/value pairs to a set of intermediate key/value pairs. Maps are the individual tasks that transform input records into intermediate records.
- The transformed intermediate records do not need to be of the same type as the input records. A given input pair may map to zero or many output pairs.
- The Hadoop Map Reduce framework spawns one map task for each Input Split generated by the Input Format for the job.
- Overall, mapper implementations are passed to the job via Job. Set Mapper Class (Class) method. The framework then calls map (Writable Comparable, Writable, and Context) for each key/value pair in the Input Split for that task. Applications can then override the cleanup (Context) method to perform any required cleanup.
- Output pairs do not need to be of the same types as input pairs. A given input pair may map to zero or many output pairs. Output pairs are collected with calls to context. Write (writable Comparable, Writable).
- Applications can use the Counter to report its statistics.
- All intermediate values associated with a given output key are subsequently grouped by the framework, and passed to the Reducer(s) to determine the final output. Users can control the grouping by specifying a Comparator via Job. Set.
- The Mapper outputs are sorted and then partitioned per Reducer. The total number of partitions is the same as the number of reduce tasks for the job. Users can control which keys (and hence records) go to which Reducer by implementing a custom Partitioned.

- Users can optionally specify a combiner, via Job.setCombinerClass (Class), to perform local aggregation of the intermediate outputs, which helps to cut down the amount of data transferred from the Mapper to the Reducer.
- The intermediate, sorted outputs are always stored in a simple (key-Len, key, value-Len, value) format. Applications can control if, and how, the intermediate outputs are to be compressed and the Compression Codec to be used via the Configuration.

5.6.2 How Many Maps?

The number of maps is usually driven by the total size of the inputs, that is, the total number of blocks of the input files. The right level of parallelism for maps seems to be around 10-100 maps per-node, although it has been set up to 300 maps for very cpu-light map tasks. Task setup takes a while, so it is best if the maps take at least a minute to execute. Thus, if you expect 10TB of input data and have a block size of 128MB, you'll end up with 82,000 maps, unless Configuration. Set(MRJobConfig.NUM_MAPS, int) (which only provides a hint to the framework) is used to set it even higher.

5.7 Reducer

- Reducer reduces a set of intermediate values which share a key to a smaller set of values.
- The number of reduces for the job is set by the user via Job.setNumReduceTasks (int).
- Overall, Reducer implementations are passed the Job for the job via the Job.setReducerClass (Class) method and can override it to initialize them. The framework then calls reduce(writable Comparable, Enterable<Writable>, Context) method for each <key, (list of values)> pair in the grouped inputs. Applications can then override the cleanup (Context) method to perform any required cleanup.

5.7.1 Reducer has 2 primary phases: shuffle, sort and reduce

Shuffle: Input to the Reducer is the sorted output of the mappers. In this phase the framework fetches the relevant partition of the output of all the mappers, via HTTP.

Sort: The framework groups Reducer inputs by keys (since different mappers may have output the same key) in this stage.The shuffle and sort phases occur simultaneously; while map-outputs are being fetched they are merged.

Reduce: In this phase the reduce (Writable Comparable, enterable<Writable>, and Context) method is called for each <key, (list of values)> pair in the grouped inputs. The output of the reduce task is typically written to the File System via Context. Write (Writable Comparable, Writable).

5.8 How Many Reduces?

The right number of reduces seems to be 0.95 or 1.75 multiplied by (<no. of nodes> * <no. of maximum containers per node>).

With 0.95 all of the reduces can launch immediately and start transferring map outputs as the maps finish. With 1.75 the faster nodes will finish their first round of reduces and launch a second wave of reduces doing a much better job of load balancing.Increasing the number of reduces increases the framework overhead, but increases load balancing and lowers the cost of failures.The scaling factors above are slightly less than whole numbers to reserve a few reduce slots in the framework for speculative-tasks and failed tasks.

5.8.1 Reducer NONE

It is legal to set the number of reduce-tasks to *zero* if no reduction is desired. In this case the outputs of the map-tasks go directly to the File System, into the output path set by File Output Format set Output Path(Job, Path). The framework does not sort the map-outputs before writing them out to the File System.

5.8.2 Partitioner

Partitioner partitions the key space. Partitioner controls the partitioning of the keys of the intermediate map-outputs. The key (or a subset of the key) is used to derive the partition, typically by a hash function. The total number of partitions is the same as the number of reduce tasks for the job. Hence this controls which of the m reduce tasks the intermediate key (and hence the record) is sent to for reduction.

5.8.3 Counter

Counter is a facility for Map Reduce applications to report its statistics. Mapper and Reducer implementations can use the Counter to report statistics. Hadoop Map Reduce comes bundled with a library of generally useful mappers, reducers, and practitioners.

5.9 Task Implementation and Environment

The MR App Master executes the Mapper/Reducer *task* as a child process in a separate jvm. The child-task inherits the environment of the parent MR App Master. The user can specify additional options to the child jvm via the map reduce.{map reduce}.java. Opts and configuration parameter in the Job such as non-standard paths for the run-time linker to search. If the map reduce.{map reduce}.java. Opts parameters contains the symbol it is interpolated with value of task id of the Map Reduce task. Here is an example with multiple arguments and substitutions, showing jvm GC logging, and start of a password less JVM JMX agent so that it can connect with console and the likes to watch child memory, threads and get thread dumps. It also sets the maximum heap-size of the map and reduces child jvm to 512MB & 1024MB respectively. It also adds an additional path to the java.library.path of the child-jvm.

5.9.1 Memory Management

Users/admins can also specify the maximum virtual memory of the launched child-task, and any sub-process it launches recursively, using map reduce. {map reduce}.memory. The value for map reduce.{map reduce}.memory.mb should be specified in megabytes (MB). And also the value must be greater than or equal to the -Xmx passed to JavaVM, else the VM might not start. The memory available to some parts of the framework is also configurable. In map and reduce tasks, performance may be influenced by adjusting parameters influencing the concurrency of operations and the frequency with which data will hit disk. Monitoring the file system counters for a job- particularly relative to byte counts from the map and into the reduce is invaluable to the tuning of these parameters.

5.9.2 Map Parameters

A record emitted from a map will be serialized into a buffer and metadata will be stored into accounting buffers. As described in the following options, when either the serialization buffer or the metadata exceed a threshold, the contents of the buffers will be sorted and written to disk in the background while the map continues to output records. If either buffer fills completely while the spill is in progress, the map thread will block. When the map is finished, any remaining records are written to disk and all on-disk segments are merged into a single file. Minimizing the number of spills to disk can decrease map time, but a larger buffer also decreases the memory available to the mapper.

5.10 Discussion

To protect the Hadoop environment, authentication, authorization, and accounting policies for accessing and using data stored in the Hadoop clusters must be implemented. If one node of Hadoop cluster is compromised then there are chances of any type of attack and data theft is possible. Securing the data in transit as well as data at rest is required. Securing Hadoop not only involves securing access to Hadoop and securing the stored data (data at rest) but also the whole gamut of security that all IT operations use, such as network security and operating system security.

Various open source communities, IT development organizations and research institutes are working together to improve the Hadoop infrastructure, tools, and services. Big Data innovations are shared through open-source platforms which are helpful to promote Big Data technologies.

However, users facing difficulties due various versions of modules from different sources combined into a Hadoop framework. Because each Hadoop module has its own curve of maturity, there is a risk of version incompatibility inside the Hadoop framework. In addition, the integration of different modules of different vendors with different technologies on a single platform increases security risks. However, most of the time, the combination of technologies from different sources may bring hidden risks that are neither fully investigated nor tested. To overcome this issue, many IT products/solution vendors such as IBM, Cloudera, MapR, and Horton works have developed their own modules and packaged them into core Apache Hadoop API and delivered improvised Hadoop distribution to the market. One of the goals is to ensure compatibility, security, and performance of all combined modules. Currently, the Hadoop system and different distributions use more than one security solutions for securing data and computation, as mentioned in the previous section, such as Apache Ranger, Apache Sentry, and Apache Knox. Each new module has its own set of vulnerabilities. If the system is needed to be flexible and interpretable, then automatically system will be vulnerable. Hadoop requires single security solution so that vulnerability can reduce that leads to attacks due to third party software's and due to different modules used in Hadoop. The results of our the experiment shows the impact of attacks on performance as well as security breach possible due to vulnerabilities. Before implementing any module in Hadoop environment, risk management study of system and defense in depth is recommended.

5.11 Conclusion

Hadoop and Cassandra are both very scalable systems! But as often stated scalability does not solve performance efficiency issues and as such neither of these systems is immune to such problems, nor to simple misuse.

Some of the prevalent performance problems are very specific to these systems and we have not seen them in more traditional systems. Other issues are not really new, except for fact that they now occur in systems that are tremendously more distributed and bigger than before. The very scalability and size makes these problems harder to diagnose (especially in Hadoop) while often having a very high impact (as in bringing down a Cassandra cluster). Performance experts can rejoice, they will have a job for a long time to come.Traditional enterprise security products implement security controls, but are still insufficient for holistically addressing the security challenges introduced by Big Data. To address the problems posed by data aggregation, organizations must find new ways to safeguard their critical tools, techniques, and procedures used to acquire, maintain, and analyze data of particular concern to improve the robustness of its security infrastructure, as most commercially available security software have access to all real and derived data available on the network. The add-on security features provided by the third party are not useful in the Big Data environment. Furthermore, analysts and other users of information technology need more effective security training that will help them understand the specific threats they face, that how their security choices will impact the outcomes, and how to reconcile security objectives with mission objectives. Finally, system designers need to consider the security implications affecting user-facing tools. Users would benefit from features, giving them security-relevant feedback throughout their workflows, rather than using a separate security tool for forensic validation. Measuring the impact of these “little” security mechanisms may prove challenging and problematic. However, field studies such as those cited in this paper suggest that they can have a large impact on scaling security in a manner that paces with the scale of data that they protect. Thus, there is a need to upgrade the complete Hadoop system released with all security features without installing and to configure it separately.

5.12 Future Research

As large scale of data from different resources to analysis is incapable execution in Hadoop Map Reduce was remark. Our proposed Benefit to manage tasks in a benefit-aware technique and expected to improve the efficiency. Finally various algorithms which take into consideration into the real-time network are proposed to improve the performance caused by hadoop. They Considering the above mentioned advantages and limitations, Apache server was able to analyze data in few seconds. Alternative processing frameworks there’s still a lot of tooling for Map Reduce. Comparison for this analysis shows that yarn is a very strong using in-memory processing and fast. Observing that yarn ability to perform batch processing, streaming, and machine learning. It is a framework for a processing large number data. This technology is used to reduce the concurrency and reduced of data.

Also there are some plans in our future research firstly, distributed hadoop cluster is planned to be realized on several terminal instead of virtual machine on just one work station for hadoop-SPARK TECHNOLOGY, we will further the study on programming and algorithm design map reduce to expanding hadoop-SPARK TECHNOLOGY spatial analysis function. And we will make deeper mining of floating car data features and the crowd flow patterns.

References

1. Jeffrey Dean and Sanjay Ghemawa, “Map Reduce: Simplified Data Processing on Large Clusters”, Communications of the ACM, Jan 2008
2. A white paper, “Aggregations and analysis on Big Data using hadoop ecosystem”,2014
3. Jimmy Lin and Chris Dyer, “Data-Intensive Text Processing with Map Reduce”, P. 18-38, April 2010.
4. Economist Intelligence Unit: the deciding factor: Big data hadoop & Decision Making. In :Cap Gemini Report, pp. 1-22(2012)
5. Dr. Siddaraju, C. L. Sowmya, K. Rashmi and M. Rahul, “Efficient Analysis of Big Data Using Map Reduce Framework”, International Journal of Recent Development in Engineering and Technology, Vol. 2, June 2014.
6. Dominique A. Hager, “Hadoop Design, Architecture & Map Reduce Performance”.
7. C. Lynch, Big data: How do your data grow?, Nature, 457 (2008), pp.28-30.
8. C. Yang, W. Lin, and M. Liu, "A Novel Triple Encryption Scheme for Hadoop-Based Cloud Data Security," in Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on, 2013, pp. 437-442.
9. C. L. Philip, Q. Chen and C. Y. Zhang, Data intensive applications, challenges, techniques and technologies: A survey on big data, Information Sciences, 275 (2014), pp.314- 347.
10. V.Shukla, "Hadoop Security Today & Tomorrow," ed: Horton works Inc., 2014. (2014-06-13). Knox Gateway Available: “shuffle performance in apache yarn”.
11. X. Zhang, "Secure Your Hadoop Cluster with Apache Sentry," ed: Cloudera, April 07, 2014.
12. Ryan Connaught on, Kevin. Bowyer and Patrick Flynn. Fusion of face and iris biometrics. In Mark J. Burge and Kevin W Bowyer, editors, Handbook of Iris Recognition, Advances in Computer Vision and Pattern Recognition, pages 219{237. Springer London, 2013.
13. T Co. Multimodal biometric identification for large user population using fingerprint, face and iris recognition. In 34th Applied Imagery and Pattern Recognition Workshop (AIPR), pages 223-228, 2005.
14. P.A. Johnson, B. Tan, and S. Shockers. Multimodal fusion vulnerability tonon-zero e_ort (spoof) imposters. In IEEE International Workshop on Information Forensics and Security (WIFS), pages 1-5, Dec 2010.
15. Kevin W. Bowyer, Karen Hollingsworth, and Patrick J. Flynn. Image understanding for iris biometrics: A survey. Computer Vision and Image Under-standing 2008.
16. A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A Ross. Biometrics: A grand challenge. In 17th International Conference on Pattern Recognition (ICPR), volume 2, pages 935-942. IEEE, 2004.

17. John Daugman and Cathryn Downing. Epigenetic randomness, complexity and singularity of human iris patterns. Royal Society London Biologica Sciences, 268(1477):1737-1740, 2001.
18. A W Kin Kong, D. Zhang, and G. Lu. A study of identical twins palmprints for personal veri_cation. Pattern Recognition, 39(11):2149-2156, 2006.
19. G.S. Badrinath and Phalguni Gupta. Palm print based recognition system using phase deference information. Future Generation Computer Systems, 28:287{305, 2010.
20. G. S. Badrinath and P Gupta. Stockwell transform based palm print recognition. Applied Soft Computing, 2011.

Publication Certificate

“Improving Large Volume of Data Processing Using Hadoop” authored by **‘Suyogita Singh,Sameer Awasthi’** was reviewed by experts in this research area are accpected by the board of ‘Blue eyes Intelligent Enigneering and science Publication which has Published in “International Journal of Information and Computing Science” ISSN NO. 0972-1347(ONLINE), Volume-8 Issues-6, May2020.Page No.:9948-998. It will available at:
https://www.ijrte.org/diwnload/volume-8-issue-6/.

Curriculum Vitae

SUYOGITA SINGH

Contact no. 7985500829, 9415047714

Email: Suyogitasingh060795@gmail.com

House No. 137 Vill. Narayanpur, Sultanpur

CAREER OBJECTIVE

To bring out the best of my skills as an employee in well know organization, and work with full commitment and dedication for the growth and progress of the organization as well as myself.

TRAINING AND DESSERTATION

- C language training from **NIIT, Lucknow**
- JAVA And ADVANCE JAVA training from **NIIT Lucknow**
- Oracle training from **NIIT Lucknow**
- Participated in cyber security Workshop Organized Innovative ideas InfoTech society

EDUCATIONAL ATTAINMENT

COURSE	**COLLEGE**	**UNIVERSITY/ BOARD**	**PERCANTAGE**	**YEAR**	**STATUS**
B.tech	AMITY UNIVERSITY	AMITY UNIVERSITY	8.1 CGPA	2014-2018	**COMPLETED**
M.tech	BBD UNIVERSITY	BBD UNIVERSITY	9.0 SGPA	2018-2020	**COMPLETED**
10^{TH}	K.N.I.C.E	C.B.S.E	8.0 CGPA	2011	**COMPLETED**
12^{TH}	K.N.I.C.E	C.B.S.E	79%	2014	**COMPLETED**

Key Technologies:

Language: C, Netbeans

Operating System: Windows 7,UNIX, LINUX

HOBBIES

- Learning about the Informational technology
- Content writing
- Photography and Photo editing.

PERSONAL DETAILS

Father's Name: Yogesh Pratap Singh

Date of Birth: 06-07-1995

Gender: Female

Language Proficiency: Hindi, English& German

Marital Status: Single

Nationality: Indian

DECLARATION

I hereby declare that above information's are true in best of my knowledge and belief.

DATE: 20-07.20

(SUYOGITA SINGH)

BBD UNIVERSITY

BABU BANARASI DAS UNIVERSITY, LUCKNOW
CERTIFICATE OF THESIS SUBMISSION FOR EVALUATION
(Submit in Duplicate)

1. Name : Suyogita Singh
2. Enrollment No. : 1180449005
3. Thesis title: Improving Large Volume of Data Processing Using Hadoop
4. Degree for which the thesis is submitted: **M.Tech(CSE)**
5. Faculty of the University to which the thesis is submitted: **Asst. Prof. Mr. Sameer Awasthi**
6. Thesis Preparation Guide was referred to for preparing the thesis. ☐YES☐ NO
7. Specifications regarding thesis format have been closely followed. ☐YES☐ NO
8. The contents of the thesis have been organized based on the guidelines. ☐YES☐ NO
9. The thesis has been prepared without resorting to plagiarism. ☐YES ☐NO
10. All sources used have been cited appropriately. ☐YES☐NO
11. The thesis has not been submitted elsewhere for a degree. ☐YES ☐NO
12. Submitted 2 spiral bound copies plus one CD. ☐YES ☐NO

Suyogita

Name: Suyogita Singh
Roll No: 1180449005
Enrollment No: 11804490804

BABU BANARASI DAS UNIVERSITY, LUCKNOW
CERTIFICATE OF FINALTHESIS SUBMISSION
(To be submitted in duplicate)

1. Name: Suyogita Singh
2. Enrollment No. : 11804490804
3. Thesis title: Improving Large Volume of Data Processing Using Hadoop
4. Degree for which the thesis is submitted: M.Tech (CSE)
5. School (of the University to which the thesis is submitted): Babu Banarasi Das University, Lucknow

6. Thesis Preparation Guide was referred to for preparing the thesis.	☐YES ☐NO
7. Specifications regarding thesis format have been closely followed.	☐YES ☐NO
8. The contents of the thesis have been organized based on the Guidelines.	☐YES ☐NO
9. The thesis has been prepared without resorting to plagiarism.	☐YES ☐NO
10. All sources used have been cited appropriately.	☐YES ☐NO
11. The thesis has not been submitted elsewhere for a degree.	☐YES ☐NO
12. All the corrections have been incorporated.	☐YES ☐NO
13. Submitted 4 hard bound copies plus one CD.	☐YES ☐NO

(Signature(s) of the Supervisor(s))
Name(s):Mr. Sameer Awasthi

Suyogita
Name:Suyogita Singh
Roll No: 1180449005
Enrollment No: 11804490804