A COMPARATIVE STUDY ON DIFFERENT BIG DATA TOOLS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Sifat Ibtisum

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

September 2020

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A COMPARATIVE STUDY ON DIFFERENT BIG DATA TOOLS

By

Sifat Ibtisum

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Kendall Nygard
Chair

Pratap Kotala

Chad Ulven

Approved:

October 15, 2020
Date

Simone Ludwig
Department Chair

ABSTRACT

Big data has long been the topic of fascination for computer science enthusiasts around the world, and has gained even more prominence in recent times with the continuous explosion of data resulting from the likes of social media and the quest for tech giants to gain access to deeper analysis. This paper discusses various tools in big data technology and conducts a comparison among them. Different tools namely Sqoop, Apache Flume, Apache Kafka, Hive, Spark and many more are included. Various datasets are used for the experiment and a comparative study is made to figure out which tool works faster and more efficiently over the others, and explains the reason behind this.

ACKNOWLEDGMENTS

First and foremost, praises and thanks to the almighty Allah, for his blessing throughout everything and finely being able to finish my paper which is the last requirement of my Master's program.

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Kendall E Nygard, Professor and Head of the Department of Computer Science at North Dakota State University for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity, and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy, and a great sense of humor. Again, due his absence Dr. Simone Ludwig took over as my chair, and the way she has guided me is really press worthy. Her sincerity, hard work and punctuality was the key reason why I am being able finish my paper so early. I also would like to extend my gratitude to my co-supervisors Dr. Ulven and Dr. Kotala for extending their support in the best possible way.

My heartiest thanks to the person, who brought me here is my mother Dr. Ferdouse Islam. My dad expired when I was 13, since then it was my mom who did everything possible to bring me where I am now. Without her unconditional support, I could never be the person who I am today. I still can remember those early days when all the odds were against us but it was she, who single-handedly made things possible. I can't put into words, how proud, honored and blessed I am to be her son. May Almighty grant her a healthy and longer life. Moreover, my little

sister Prioty, who made countless sacrifices so that I can achieve my dreams. Wish I could ever put into words, how thankful I am to you, my little angle.

I also would like to extend my gratitude to my families and friends who believed in my dreams and helped me to progress. Among my families, Jes and Mr. Rony to whom I am really thankful for their exceptional support towards my family. And among my friends, Mr. Rafsan and Mr. Opu are the two people who acted like my elder brothers and did everything possibly they can so that I can achieve my dream. I am truly honored to have you guys in my life.

Lastly, I want to thank everyone in the computer science Dept, graduate school and international office of North Dakota State University to guide me throughout this journey and last but not the least each member of NDSU for making my journey so exceptional and memorable.

DEDICATION

I dedicate this work to the most valuable person of my life, my mom.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Almost quintillion bytes of data is being generated each day. Around 90% of the total data is created in just last two years only. Data at this point accumulates from everywhere, including climate information gathering sensors, social media websites, videos and pictures, transaction records (e.g. banking records), and cell phone calls, GPS signals, etc.

In this paper, all the major tools of big data are introduced. The prime focus of this paper is to make a comparison among all the tools in big data technology, explaining their pros and cons. Then, run different experiments using various data sets of different sizes to validate the study and to explain the results. Graphical representation is being used to provide visual presentation of how one tool is outperforming the others for certain types of data.

The ever-increasing use of the internet, sensors and heavy machines at a very high rate with sheer volume, velocity, variety and veracity, the data is termed as Big Data. Data is everywhere, in every industry, in the form of numbers, images, videos and text. As data continues to grow, it becomes difficult for the computing system to manage big data due to immense speed and volume at which it is generated. As the data is enormous and complex, the data is stored in a distributed architecture file system. Analyzing the complex data is a risky and time-consuming task as it contains big distributed file systems, which should be fault tolerant, flexible, and scalable. The process of capturing or collecting big data is known as 'datafication'. Big data is 'datafied' so that it can be used productively. Big Data cannot be made useful by simply organizing it, rather the data's usefulness lies in determining what we can do with it [1].

Big Data is **engendered** by almost everything around us, like social network, government, healthcare, education at an alarming volume with high velocity and variety. To pull out meaningful assessment from this enormous data, it is necessary to do best possible processing control, analytical potential and skills.

As big data contains data which is big in size, the selection of right data within the larger data set to analyze the whole data should be appropriate. Predictive analytics and data mining solutions for the enterprises are currently available from a number of companies, like Predictive analytics Suit, IBM SPSS Statistics, Microsoft Dynamics CRM Analytics Foundation [1]. Software on big data platforms and big data analytics focus on giving efficient analysis on data on enormously big datasets. The industries like banking, automobiles, healthcare, telecom, government, transportation and travel will have major impact through Big Data analytics (BDA). Its perspectives are to provide assistance to industries to extract data into high-quality information for in-depth approach into their organizations status [1].

The expansion of data by no means can be stopped or it cannot be restricted anyhow. IDC Digital Universe in its Study, published in year 2011 stated, nearly 130 Exabytes of data was produced and stored in 2005. This quantity grew severely to 1,227 Exabytes in 2010 and it was projected to produce at 45.2% to 7,910 Exabytes in year 2015. This data can do wonder by extracting the buried assets of information that reside inside it. In the year 2004 “Google” introduced MapReduce to the world, which later laid foundation to Hadoop and other related technologies (methods). Goal of Hadoop was to index the entire WWW (World Wide Web) and today the open-source Hadoop technology is being used many organizations to munch through large volumes of data [2].

The global phenomena of using Big Data to gain business value and competitive advantage will only continue to grow as will the opportunities associated with it. As per MGI and McKinsey's Business Technology Office research, the utilization of enormous Data is most likely to become a key basis of competition for individual firms for success and growth and strengthening consumer surplus, production growth and innovation. The means of using Big Data is by selecting appropriate Big Data analytics platform and the tools which is very critical for an organization [3].

This paper is a handbook of Big data. It provides all the necessary detail of big data including its characteristics and types, its 5 Vs, various structures of data, why processing big data is so important, the applications of big data, different tools used in big data for analyzing the dataset. Moreover, various experiments were performed using various datasets to prove which tools are better than the other. A comparative study of the data interpretation is shown for the better understanding of the topic.

1.1. Characteristics of Big Data

The origin of large data sets began in the 1960 and 70s with the introduction of first data center and the development of relational database.

Around 2005, with the introduction of various social media like Facebook, YouTube and other online services, people started to release how much data is being generated by users. And in the same year an open source framework 'Hadoop' was created specifically to store and analyze the big data sets. NoSQL database that provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases as was introduced around the same time.

Hadoop made a breakthrough as an open-source framework since it made big data easier to work with and cheaper to store. In next few years, the volume of big data has skyrocketed. And at this point, it's not just the human who generates data rather the devices as well which gives us many crucial insides.

Now with the help of IoT (Internet of Things), more devices including phone, watch. Fan, light are connected with internet. They provide lots of detail of the performance and user experience which led us to understand user behavior and patter. Introduction of machine learning has also produced lots of data.

While big data has come so far, its usefulness is only just increasing. Cloud computing has expanded big data possibilities even further. Cloud offers truly elastic scalability, where developers can simply spin up ad hoc clusters to test a subset of data.

1.2. Benefits of Big Data and Data Analytics

- Big data makes it possible for you to gain more complete answers because you have more information.
- More complete answers mean more confidence in the data—which means a completely different approach to tackling problems.

Big data gives you new insights that open new opportunities and business models. Getting started involves three key actions:

a. Integrate: Big data brings together data from many disparate sources and applications. Traditional data integration mechanisms, such as ETL (extract, transform, and load) generally are not up to the task. It requires new strategies and technologies to analyze big data sets at terabyte, or even petabyte, scale. During integration, you need to bring in the data, process it, and make sure it's formatted and available in a form that your business analysts can get started with.

b. Manage: Big data requires storage. Your storage solution can be in the cloud, on premises, or both. You can store your data in any form you want and bring your desired processing requirements and necessary process engines to those data sets on an on-demand basis. Many people choose their storage solution according to where their data is currently residing. The cloud is gradually gaining popularity because it supports your current compute requirements and enables you to spin up resources as needed.

c. Analyze: Your investment in big data pays off when you analyze and act on your data. Get new clarity with a visual analysis of your varied data sets. Explore the data further to make new discoveries. Share your findings with others. Build data models with machine learning and artificial intelligence. Put your data to work.

1.3. What is Big Data

Big Data is the term for a collection of data sets so large and complex that it becomes difficult to process using conventional data mining techniques and tools. The overall goal of the big data analytics is to extract useful information from a huge data set and transform it into an understandable structure for further use. The major processes of big data include capture, curation, storage, search, sharing, transfer, analysis, and visualization.

Recently the importance of this field has attracted enormous attention because it gives businesses useful information and better insight of both structured and unstructured data, which may lead to better informed decision-making. In a business context, big data analytics is the process of examining “big data” sets to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. Today’s advances in technology combined with the recent developments in data analytics algorithms and approaches have made it possible for organizations to take advantage big data analytics. Some of the major issues in applying big data analytics successfully include data quality, storage, visualization and processing. Some business examples of big data are social media content, mobile phone details, transactional data, health records, financial documents, Internet of things and weather information.

There are many definitions of Big Data framed differently in the past by various researchers, but all of them revolve around the five characteristics of Big Data. These 5 V's of Big Data are:

1) Variety: The first characteristic of Big Data is Variety, which addresses the various sources which are generating this Big Data. They are classified into three categories as:
   a) Structured Data: Structured data concerns all data which can be stored in table with rows and columns. These data are considered to be the most organized data, but it accounts for only 5-10% of the total data available.
   b) Semi structured data: Semi-structured data is the information that does not reside in tables but they possess some properties which make them convertible into structured data. These are the data coming from web server logs, XML documents etc. Comparatively less organized than structured data, they also make only 5-10% of data available.
   c) Unstructured data: Unstructured data constitutes the biggest source of Big Data that is 80 - 90%. It includes data in the form of text, images, video, voices, web pages, emails, word documents and all other multimedia content. These data are very difficult to store into database. These types of data are both machine and human generated just like structured and semi structured data.
2) Volume: Volume is the characteristic which makes Data as Big Data. It denotes to the large amount of data which is generating in every second. The range of data has highly

increased, crossing the range of terabytes to Peta, Exa and now till Zeta bytes. Big data can be measured in the terms of:

- Records per Area
- Transactions
- Table

3) Velocity: Data is coming from multiple sources in huge amounts, as explained earlier. Also, Velocity is one of the characteristics of Big Data which talks about the high data rate at which it is being generated. Various applications based on data rate are:

- Batch: Batch means running the query in a scheduled and sequential way without any intervention. Execution is on a batch of input.

  Real Time: Real time data is defined as the information which is delivered immediately after its collection. There is no delay in the timeliness of information provided.

  Interactive means executing the tasks which require frequent user interaction.

  Streaming: The method of processing the data as it comes in is called streaming. The insight into the data is required as it arrives.

4) Value: It is necessary to fetch meaningful information or patterns from this huge amount of Big Data which can be used for analysis or determining results on application of queries. Thus, Value is the characteristic which denotes fetching meaning from Big Data. The value can be extracted from Big Data as:

- Statistical
- Events
- Correlation

- Hypothetical

5) Veracity: The fifth V of Big data ensures the correctness and accuracy of information. When dealing with Big Data, along with maintaining its privacy and security, it is also important to take care of Data quality, data governance and metadata management. Factors which should be considered are:

- Trustworthiness
- Authenticity
- Accountability
- Availability

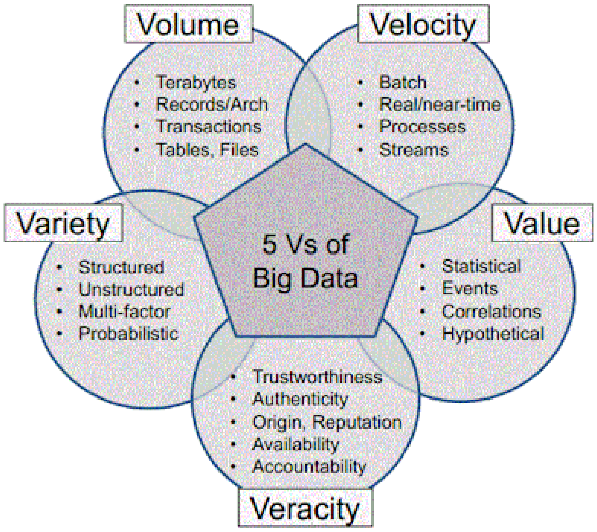


Fig 1: 5 Vs of Big Data [33]

1.4. Importance of Big Data Processing

Big data processing is the method of probing big data to uncover hidden patterns, correlations and other useful information that can be used to make improved decisions. With big data analytics, data scientists and others can analyze massive volumes of data that usual analytics and business intelligence solutions cannot tap. Consider that your business could build up billions of rows of data with millions of data combinations in numerous data stores and plentiful formats. High-performance analytics is essential to process that much data in order to outline out what's significant and what is not. For most organizations, big data analysis is defied. Consider the total volume of data and the unlike formats of the data that is composed transversely the entire business and the many types of data can be pooled, contrasted and analyzed to gather patterns and other valuable business information.

There are generally four approaches to data analytics, and each of them comes under either reactive or proactive category:

- Reactive (business intelligence). In this category, business intelligence provides pattern business reports, ad hoc reports, OLAP and even gives alerts and notifications that are based on analytics. Here ad hoc analysis checks at the static past, which has its reason in a limited number of conditions.
- Reactive – big data BI. When reporting pulls from huge data sets, we can say this is performing big data BI. But decisions based on these two methods are still reactionary.
- Proactive (big analytics). Creation of forward looking, proactive decisions needs proactive big analytics similar to optimization, predictive modelling, text mining and forecasting. They allow us to recognize trends, mark weaknesses and determine circumstances for creation of decisions about the future. But though it is proactive, big analytics can’t still be performed on big data since traditional storage techniques and processing times cannot stay up.
- Proactive (big data analytics). Using big data analytics, we can mine only the appropriate information from mountains of, and then analyze it to convert our business decisions for the better future. Flattering proactive with big data analytics is not a one-shot attempt; it

is a culture change, a new method of acquisition by freeing our analysts and decision creators to gather the future with appropriate knowledge and insight.

It becomes very difficult for the traditional data analysis, processing and storing tools to deal with the five characteristics of data simultaneously. Since big data is a recent upcoming technology in the market which can bring huge benefits to the business organizations, it becomes necessary that various challenges and issues related to it must be taken care of and resolved [5].

1.5. Application of Big Data

In today's world the application of Big data is at everywhere, here is the list of biggest contributors in generating such huge bulk of data:

- Healthcare: Healthcare industry has now shifted from single-physician offices to multi-provider groups by digitizing, combining and making effective use of data. In many cases, the multi-provider groups have shifted to large hospital network as well, where there is accountable care unit stands to realize significant benefits and transparency [2]. One potential benefit of big data could be detecting disease at earlier stages when they can be treated more easily and effectively. Managing the health portal to monitor specific individuals and detecting health care fraud more quickly and efficiently. Certain prediction or estimation can be made based on vast historical data, such as length of stay (LOS), patients who will choose elective surgery; patients who likely will not benefit from surgery; complications; patients at risk for medical complications; patients at risk for sepsis, MRSA, C. difficile, or other hospital-acquired illness; illness/disease progression; patients at risk for advancement in disease states; causal factors of illness/disease progression; and possible co-morbid conditions (EMC Consulting). Mckinsey estimates that big data analytics are capable of saving $300 billion in saving per year in U.S. Clinical operations and R&D are two of the largest areas for potential savings with $165 billion and $108 billion in waste respectively [3].

- Public sector administration: As of today, there are no broad implementations of big data in the public sector. Compared to other sectors, the public sector has not been traditionally using data mining technologies intensively. However, there is a growing interest in the public sector on the potentials of big data for improvement in the current financial environment.

Some examples of the global growing awareness are the Joint Industry/Government Task Force to drive development of big data in Ireland, announced by the Irish Minister for Jobs, Enterprise and Innovation in June 2013 (Government of Ireland 2013), or the announcement made by the Obama administration (The White House 2012), on the "Big Data Research and Development Initiative" where six Federal departments and agencies announce more than $200 million in new commitments to greatly improve the tools and techniques needed to access, organize, and glean discoveries from huge volumes of digital data [6].

- Retail: A lot of people are under the impression that great marketing is an art, but of late, big data has introduced a scientific element to marketing campaigns. Smart marketers are now relying on data more than ever to inform, test, and devise their strategies. And though data and analytics will never replace the creative minds behind the best marketing campaigns, it can provide the marketers with the tools to help perform better. Consumers have 24-hour access to abundant product information, which has revolutionized the retail sector. With digital technology becoming ubiquitous, shoppers can make informed decisions using online data and content to discover, compare, and buy products from anywhere and at any time. For brands and retailers, information is also a game-changer.

Retail data analytics can help companies stay abreast of the shopping trends by applying customer analytics to uncover, interpret, and act on meaningful data insights, including online shopper and in-store patterns. The retailers—both offline and online—are adopting the data-first strategy towards understanding the buying behavior of their customers, mapping them to products, and planning marketing strategies to sell their products to register increased profits. Today, retailers attempt to find innovative ways to draw insights from the ever-increasing amount of structured and unstructured information available about their consumer's behavior. Big Data analytics is now being applied at every step of the retail process - right from predicting the popular products to identifying the customers who are likely to be interested in these products and what to sell them next.

- Manufacturing: The manufacturing industry is currently in the midst of a data-driven revolution, which promises to transform traditional manufacturing facilities into highly optimized smart manufacturing facilities. These smart facilities are focused on creating manufacturing intelligence from real-time data to support accurate and timely decision-making that can have a positive impact across the entire organization. To realize these efficiencies emerging technologies such as Internet of Things (IoT) and Cyber Physical Systems (CPS) will be embedded in physical processes to measure and monitor real-time data from across the factory, which will ultimately give rise to unprecedented levels of data production. Therefore, manufacturing facilities must be able to manage the demands of exponential increase in data production, as well as possessing the analytical techniques needed to extract meaning from these large datasets. More specifically, organizations must be able to work with big data technologies to meet the demands of smart manufacturing. However, as big data is a relatively new phenomenon and potential applications to manufacturing activities are wide-reaching and diverse, there has been an obvious lack of secondary research undertaken in the area. Without secondary research, it is difficult for researchers to identify gaps in the field, as well as aligning their work with other researchers to develop strong research themes. In this study, we use the formal research methodology of systematic mapping to provide a breadth-first review of big data technologies in manufacturing [6].

- Personal location data: Geospatial data has always been big data. In these days, big data analytics for geospatial data is receiving considerable attention to allow users to analyze huge amounts of geospatial data. Geospatial big data typically refers to spatial data sets exceeding capacity of current computing systems. McKinsey Global Institute says that the pool of personal location data was in the level of 1 PB in 2009 and is growing at a rate of 20% per year [1]. This estimation did not include the data from RFID sensors and those stored in private archives. According to the estimation by United Nations Initiative on Global Geospatial Information Management (UN-GGIM), 2.5 quintillion bytes of data is being generated every day, and a large portion of the data is location-aware. Also, in Google, about 25 PB of data is being generated per day, and a significant portion of the data falls into the realm of spatio-temporal data. This trend will be even accelerated since the world becomes more and more mobile in these days [7].
- Cost reduction - Big data technologies like Hadoop and cloud-based analytics can provide substantial cost advantages. There is a rising trend in equipping vehicle from company fleets with IOT (Internet of Thing) through with the management can keep track of the vehicle and driver's health. Research showed this practice can reduce the cost of fuel consumption and CO2 emission. Again, since companies like to go through

frequent testing of the product and the market before launching their product, analytics platforms make tests less time-consuming and there by not as expensive. Moreover, cyber-attacks can disrupt the website functionality, erode customer trust eventually destroy company status. And to stop that, companies need data analytics platforms for cybersecurity purposes that can check network traffic continually and give notification of suspicious activities. Of course, these platforms cost way less than the amount companies loose due to all the cyber-attacks.

- Faster, better decision making - Analytics has always involved attempts to improve decision making, and big data does not change that. Following the Big data analytics really makes the business managers good decision makers. Large organizations are seeking both faster and better decisions with big data, and they are finding them. Driven by the speed of Hadoop and in-memory analytics, several companies are focused on speeding up existing decisions.
- Fraud Detection - High-performance analytics is not just another technology fad. It represents a revolutionary change in the way organizations harness data. With new distributed computing options like in-memory processing on commodity hardware, businesses can have access to a flexible and scalable real-time big data analytics solution at a reasonable cost. This is sure to change the way insurance companies manage big data across their business – especially in detecting fraud.
- Fact based decision making: The term "big data" is being labelled as the next big thing as far as innovation is concerned [8]. The implications of big data for business solutions are far reaching, extending to all domains including organized retail [35]. Organizations in this sector are increasingly collecting, storing, and analyzing substantial granular data and

information – acquired through systematic processes and systems – which are essentially about products or services meant for sale (along with all tagged information), such as people (both customers and employees), and transactions (primarily the interactions and sale closings between employee-customer pairs). This data-linked practice to boost consumer purchases has seen more engagement than ever before. The standard collaboration tools include cloud services, email storage, POS data tracker, mobile devices and other similar gadgets required to conduct business and interact with suppliers, customers and other stakeholders [8]. Such applications create, receive and collect machine and/or sensor-generated data messages at very high volumes, which then drive business processes. As such, detailed insights and patterns of data can be identified, which tend to reveal, for example, a customer's tastes, purchase preferences, levels of spending; and also more importantly, calculated enticement becomes possible. For instance, if User 1 buys Item 1 and wishes for Item 2, whilst User 2 simply purchases Item 1, then it is highly likely that when strategically focused and pushed, User 2 will also end up wishing for Item 2, and all of this can be managed in real-time. To date, the extant literature and academic discussions have mostly dealt with the aspect of generating more revenues through the sale of, or adding value to, or modifying an already created Item 2. However, it is plausible that a newer Item 2 can be created from scratch, starting with ideation at the POS itself when a derivative is obtained from interacting knowledge fragments of the salesforce and the customers. That said, there is a pressing need for education, training and awareness for this, which can also be achieved during these processes. The organizations' expected objective will be in creating newer cost-effective products that not only induce a spike in customers' interest (as well as revenues) but also allow greater diversification in the portfolio of products for competitive advantage. Decision-making related to creating these cost-effective products will be supplemented by evidence from big data based. Big data helps organizations become resourceful enough to tackle diverse business challenges [8] and build organizational capability [8]. It is implied that relevant organizational knowledge can be extracted from prodigious volumes of big data available to retail organizations. Along with that, the professionals'/personnel's personal knowledge will also be at those organizations' disposal. Such knowledge may combine and recombine to give rise to an organizational process of knowledge co-creation, involving other stakeholders (customers) through the use of information communication tools (ICT) for their positive impact on SECI (socialization, externalization, combination and internalization) [9]. Furthermore, the knowledge co-creation process will be catalyzed by the inducing factors such as intention, autonomy and fluctuation. In the end, the entire process will lead to possible assistance in evidence-based decision-making. The aforementioned form of decision-making in essence allows efficient and effective decisions to be made. These decisions are likely to generate value for the business [9]. At the same time, during the entire organizational process of knowledge co-creation, the elements of relevant organizational knowledge will be defined by individual knowledge and group knowledge, while personal knowledge will be defined by explicit knowledge and tacit knowledge. In this context, knowledge will refer to knowledge of customers [owing to the nature of the selected sample]. It has been posited that no organization can exist in isolation and they will regularly interact with their environment (which itself changes dynamically), thereby dispersing data, information and knowledge by various means, building the organizations in that manner as well as helping the organizational members understand their organizations and their requirements [10] Thus, a retail organization can rely on efficient and effective evidence-based decision-making for generating business value through knowledge co-created in the organization – which carries the potential of relevant organizational knowledge (at individual and group level) as extracted from big data and personal knowledge of employees (both explicit and tacit). Evidence-based decision-making can prove to be a core competency for the organization. In order to reap the benefits, the organization will be making different attempts for advancing in the forward-looking direction [10].

- Improved customer experience: Customer sentiments are unreliable and uncertain due to subjectivity of human opinions. Statistical tools and techniques have been developed to deal with uncertainty and unreliability of big data with specified confidence levels or intervals. SAS added two additional dimensions to big data: variability and complexity. *Variability* refers to the variation in data flow rates. In addition to the increasing velocity and variety of data, data flows can fluctuate with unpredictable peaks and troughs. Unpredictable event-triggered peak data are challenging to manage with limited computing resources. On the other hand, investment in resources to meet the peak-level computing demand will be costly due to overall underutilization of the resources. Complexity refers to the number of data sources. Big data are collected from numerous data sources. Complexity makes it difficult to collect, cleanse, store, and process heterogeneous data. It is necessary to reduce the complexity with open sources, standard platforms, and real-time processing of streaming data. Oracle introduced *value* as an additional dimension of big data. Firms need to understand the importance of using big data to increase revenue, decrease operational costs, and serve customers better; at the same time, they must consider the investment cost of a big data project. Data would be low value in their original form, but data analytics will transform the data into a high-value strategic asset. IT professionals need to assess the benefits and costs of collecting and/or generating big data, choose high-value data sources, and build analytics capable of providing value-added information to managers. Social media analytics support social media content mining, usage mining, and structure mining activities. Social media analytics analyze and interpret human behaviors at social media sites, providing insights and drawing conclusions from a consumer's interests, web browsing patterns, friend lists, sentiments, profession, and opinions. By understanding customers better using social media analytics, firms develop effective relationship marketing campaigns for targeted customer segments and tailor products and services to customers' needs and interests. For example, major U.S. banks analyze clients' comments on social media sites about their service experiences and satisfaction levels. Unlike web analytics used mainly for structured data, social media analytics are used for the analysis of data likely to be natural language, unstructured, and context dependent. The worldwide social media analytics market is growing rapidly from $1.6 billion in 2015 to an estimated $5.4 billion by 2020 at a compound annual growth rate of 27.6%. This growth is attributable to advanced analytics and the increase in the number of social media users [10]. Some social media analytics software programs are provided as cloud-based services with flexible fee options, such as monthly subscription or pay-as-you-go pricing. Social media analytics focus on two types of analysis: sentiment analysis and social network analysis. Sentiment analysis uses text analysis, natural language processing, and computational linguistics to identify and extract user sentiments or opinions from text materials. Sentiment analysis can be performed at multiple levels, such as entity level, sentence level, and document level. An entity-level analysis identifies and analyzes individual entity's opinions contained in a document. A sentence-level analysis identifies and analyzes sentiments expressed in sentences. A document-level analysis identifies and analyzes an overarching sentiment expressed in the entire document. However, sentiment analysis can be flawed. Sampling biases in the data can skew results as in situations where satisfied customers remain silent while those with more extreme positions express their opinions [10]. By exploiting big data from multiple sources, firms can deliver personalized product/service recommendations, coupons, and other promotional offers. Major retailers such as Macy's and Target use big data to analyze shoppers' preferences and sentiments and improve their shopping experience. Innovative fintech firms have already started using social media data to assess the credit risk and financing needs of potential clients and provide new types of financial products for them. Banks are analyzing big data to increase revenue, boost retention of clients, and serve clients better. U.S. Bank, a major commercial bank in the U.S., deployed data analytics that integrate data from online and offline channels and provide a unified view of clients to enhance customer relation management. As a result, the bank's lead conversion rate has improved by over 100% and clients have been able to receive more personalized experiences [10]. Harnessing big data collected from customer interactions allows firms to price appropriately and reap the rewards [10]. Sears uses big data to help set prices and give loyalty shoppers customized coupons. Sears deployed one of the largest Hadoop clusters in the retail industry and now utilizes open source technologies to keep the cost of big data low. Sears analyzes massive amounts of data about product availability in its stores to prices at other retailers to local weather conditions in order to set prices dynamically. eBay also uses open source Hadoop technology and data analytics to optimize prices and customer satisfaction. To achieve the highest price possible for items sellers place for auction, eBay examines all data related to items sold before (e.g., a relationship between video quality of auction items and bidding prices) and suggests ways to maximize results to sellers. Big data analytics can integrate data from multiple communication channels (e.g., phone, email, instant message) and assist customer service personnel in understanding the context of customer problems holistically and addressing problems quickly. Big data analytics can also be used to analyze transaction activities in real time, detect fraudulent activities, and notify clients of potential issues promptly. Insurance claim representatives can serve clients proactively based on the correlation analysis of weather data and certain types of claims submitted on stormy or snowy days. Hertz, a car rental company in the U.S., uses big data to improve customer satisfaction. Hertz gathers data on its customers from emails, text messages, and online surveys to drive operational improvement. For example, Hertz discovered that return delays were occurring during specific hours of a day at an office in Philadelphia and was able to add staff during the peak activity hours to make sure that any issues were resolved promptly. Southwest Airlines uses speech analytics to extract business intelligence from conversations between customers and the company's service personnel. The airline also uses social media analytics to delve into customers' social media data for a better understanding of customer intent and better service offerings [11].

- Improved sales: Modern-day marketing has changed drastically over the past decade and that is all due to big data. It has completely changed the business models which is more convenient to the customers and eventually brings more profit. According to a BARC research report, businesses that use big data saw a profit increase of 8 percent, and a 10 percent reduction in overall cost. There are many ways big data can be used for the business growth and advertising. Here are examples of how companies are using big data today, and how your company can use it to boost sales. How often have you looked at your Amazon recommendations and thought, "Wow, I could really use that!" Chances are, that reaction happens fairly often considering that Amazon uses big data to figure out exactly the type of products you will want to buy in the future[12].
- The retail giant -- gives its customers insight as to what factors go into determining those recommended products. In this context, Amazon cites a variety of data points to figure out what its customers want. Those factors include:

1. When customers make purchases
2. How customers rate their purchases
3. What customers with similar buying habits are purchasing
4. Obviously, the last factor is the most important one as pertains to big data. Amazon is able to correctly determine what kind of products customer wants to buy based on customers buying habits.
5. Similarly, this kind of data can be used to make predictions for your own customers. When you see a sales increase, you'll start to notice trends. For example, Amazon noticed that people who buy TVs tend to also purchase a TV mount -- which the retailer began to upsell in the hope that customers would buy them together.

Now since we are connected with each other one way or the other, an entire business can be compromised just by few keystrokes[13].

Operational risk is significantly higher in financial institutes. Scammers are constantly involved to evolve schemes to take advantage of both people and companies. As big data has evolved, however, financial institutes have realized that they can use this information to stop scam artists in their tracks.

Banks, are now using big data to monitor their transactions on a “front-to-back” business line to help eliminate fraud at all levels. They track the information about who is sending/receiving money, how often those people engage in this behavior, where they live and how much money they are sending[14].

These types of technologies not just helps banks rather any business can use them. As data is collected, trends emerge and anything that deviates from “business as usual” triggers a digital sticky note on that transaction. This makes companies to track every detail and reduce the risk of the business.

New product innovation: Big data is currently a common problem faced by many industries, and it brings grand challenges to these industries’ digitization and Informationization. Research on common problems of big data, especially on breakthroughs of core technologies, will enable industries to harness the complexity induced by data interconnection and to master uncertainties caused by redundancy and/or shortage of data. Everyone hopes to mine from big data demand-driven information, knowledge and even intelligence and ultimately taking full advantage of the big value of big data. This means that data is no longer a byproduct of the industrial sector but has become a key nexus of all aspects[15]. In this sense, the study of common problems and core technologies of big data will be the focus of the new generation of

IT and its applications. It will not only be the new engine to sustain the high growth of the information industry, but also the new tool for industries to improve their competitiveness. For example, in recent years, cloud computing has rapidly evolved from a vague concept in the beginning to a mature hot technology. Many big companies, including Google, Microsoft, Amazon, Facebook, Alibaba, Baidu, Tencent, and other IT giants, are working on cloud computing technologies and cloud-based computing services. Big data and cloud computing is seen as two sides of a coin: big data is a killer application of cloud computing, whereas cloud computing provides the IT infrastructure to big data[16]. The tightly coupled big data and cloud computing nexus are expected to change the ecosystem of Internet, and even affect the pattern of the entire information industry. Big data technologies and the corresponding fundamental research have become a research focus in academia. An emerging interdisciplinary discipline called data science [17] has been gradually coming into place. This takes big data as its research object and aims at generalizing the extraction of knowledge from data. It spans across many disciplines, including information science, mathematics, social science, network science, system science, psychology, and economics [17,18]. It employs various techniques and theories from many fields, including signal processing, probability theory, machine learning, statistical learning, computer programming, data engineering, pattern recognition, visualization, uncertainty modeling, data warehousing, and high-performance computing. Many research centers/institutes on big data have been established in recent years in different universities throughout the world (such as Tsinghua University, the University of California at Berkeley, Columbia University, New York University, Eindhoven University of Technology, and Chinese University of Hong Kong). Lots of universities and research institutes have even set up undergraduate and/or postgraduate courses on data analytics for cultivating talents, including data scientists and data engineers.

1.6. Big Data and Cloud Computing Challenges

The fact that the valuable enterprise data will reside outside the corporate firewall raises serious concerns. Some of the most common challenges are discussed below:

- Data Storage - Storing and analyzing large volumes of data that is crucial for a company to work requires a vast and complex hardware infrastructure. With the continuous growth of data, data storage device is becoming increasingly more important, and many cloud companies pursue big capacity of storage to be competitive [19].
- Data Quality - Accuracy and timely availability of data is crucial for decision-making. Big data is only helpful when an information management process is implemented to guarantee data quality [20].
- Security and Privacy - Security is one of the major concerns with big data. To make more sense from the big data, organizations would need to start integrating parts of their sensitive data into the bigger data. To do this, companies would need to start establishing security policies which are self-configurable: these policies must leverage existing trust relationships and promote data and resource sharing within the organizations, while ensuring that data analytics are optimized and not limited because of such policies. Hacking and various attacks to cloud infrastructure would affect multiple clients even if only one site is attacked. These risks can be mitigated by using security applications, encrypted file systems, data loss software, and buying security hardware to track unusual behavior across servers [21].

- Service Delivery and Billing- It is difficult to assess the costs involved due to the on-demand nature of the services. Budgeting and assessment of the cost will be very difficult unless the provider has some good and comparable benchmarks to offer. The service-level agreements (SLAs) of the provider are not adequate to guarantee the availability and scalability. Businesses will be reluctant to switch to cloud without a strong service quality guarantee [22].
- Interoperability and Portability Businesses- should have the leverage of migrating in and out of the cloud and switching providers whenever they want, and there should be no lock-in period. Cloud computing services should have the capability to integrate smoothly with the on-premise IT [23].
- Reliability and Availability- Cloud providers still lack round-the-clock service; this results in frequent outages. It is important to monitor the service being provided using internal or third-party tools. It is vital to have plans to supervise usage, SLAs, performance, robustness, and business dependency of these services [24].
- Performance and Bandwidth Cost Businesses- can save money on hardware but they must spend more for the bandwidth. This can be a low cost for smaller applications but can be significantly high for the data-intensive applications. Delivering intensive and complex data over the network requires sufficient bandwidth [25].

All these challenges should not be considered as roadblocks in the pursuit of cloud computing. It is rather important to consider these issues and the possible ways out before adopting the technology.

2. TARGET AUDIENCE OF THE BIG DATA TOOLS

Nearly everything we do in our day-to-day lives leaves a digital trail. **Big data** is the collection of all of this digital information in complex datasets; it can be analyzed to identify trends and applied to improve services. Big data has permeated professional sectors from business and tech to agriculture and health – and for good reason. When used effectively, big data can help companies better target customers, identify relationships between clients and their needs, and foster more strategic decision-making. Essentially, big data allows companies to know their customers in order to improve their marketing strategies and the customer experience [26].

2.1. Knowing your Current and Future Customers

Big data provides crucial insight about the customers, including demographics, geographic location, and how they interact with the company in real-time. Through analysis of this data, you can develop profiles for your customers to better define your audience and their interests. Perhaps most importantly, data provide the opportunity for you to learn more about your customers' attitudes and behaviors in order to predict behavior patterns and better target your customers.

In addition to gaining valuable insights into your current customers to continue meeting their needs, big data also allows you to profile future customers. Using data to identify characteristics of current valuable customers; applying predictive marketing; and using data from such sources as website analytics and social media metrics, among others, will produce further data that lends to developing new target customers [27].

2.2. Targeting your Customers

Knowing who your current and future customers are allows you to develop more personalized marketing campaigns to target customers, a strategy that is central to the account-based marketing (ABM) approach. ABM applies a focused, personalized marketing strategy with messaging that is tailored to the customers' specific needs and traits as gleaned through data analysis. Yielding higher engagement with target customers due to this personalized approach, ABM offers a higher return on investment than other B2B marketing strategies [28].

2.3. Improving the Customer Experience

Hand-in-hand with knowing and targeting your customers is then using that knowledge to improve the customer experience. Big data allows you to gather real-time information about your customers' behavior, preferences, and product reactions to better meet their needs. Such information allows you to identify patterns and improve what you offer to the customers (e.g., Amazon, Netflix, and Spotify's personal suggestions based on their users' shopping/viewing/listening history). Similarly, data can show you what's not working for your customers. If data reveal that customers aren't using a certain service or product, you can use that information to adjust your targeting strategy [29].

Improving the customer experience is not all about what's happening between customers and your company. Using the data to gain a pulse on what your competitors are doing and where they're finding success. Anything from tweets to reports can hold important insight into what your customers are looking for and help you provide the expectation that will keep them coming back.

2.4. Being Proactive

Big data holds significant potential for expanding your company by enabling you to more strategically target your customer base and improve their experience interacting with your company. However, simply gathering the data isn’t enough. You must be proactive in analyzing your data for trends and patterns and applying these insights to target customers. Big data provides a steady stream of data points; it is up to you to commit to using them [30].

3. LIST OF TOOLS USED IN BIG DATA

3.1. Sqoop

Sqoop is a tool designed to transfer data between Hadoop and relational database servers (RDMS). It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS or related Hadoop eco-systems like Hive and HBase and export from Hadoop file system to relational databases [31].

Following is the Sqoop architecture that is used to transfer data between relational database (MySql, Oracle, Postgresql, DB2) and Hadoop File System (HDFS, Hive, HBase). Fig. 2 shows the Sqoop architecture.

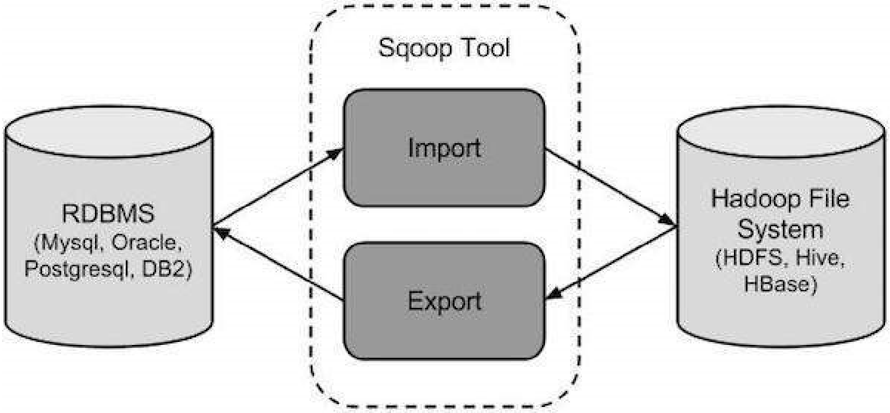


Fig 2: Sqoop architecture [2]

3.2. Apache Flume

Apache Flume is a tool / data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events etc. from various sources to a centralized data store. Flume is a highly reliable, distributed and configurable tool whose job is to copy streaming log data from various web servers to HDFS. Flume can be used to store data into any centralized store like HBase, Hive. Along with the log files, flume can be used to import huge volume of event data produced by social networking sites like Facebook, Twitter or any e-commerce websites like Amazon. Fig.3 shows the Apache Flume architecture [32].

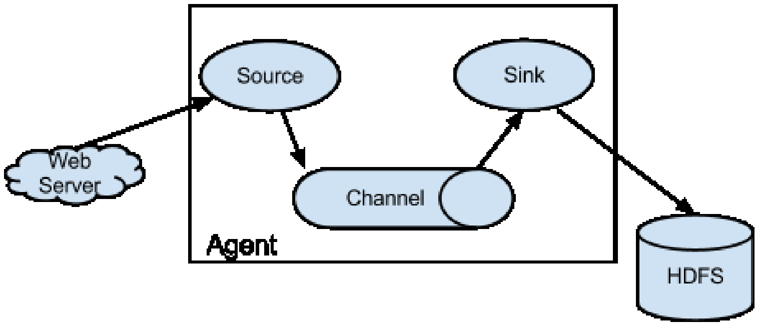


Fig 3: Flume architecture [2]

In Fig. 3, events generated by external source like webserver are consumed by Flume Data Source. The external source sends events to Flume source in a format that is recognized by the target source. Then the source receives an event and stores it into one or more channels. The channel acts as a store which keeps the event until it is by flume sink. And the channel may use the local file system to store these events. Lastly, the flume sink removes the event from a channel and store it into an external repository like HDFS [32].

3.3. Apache Kafka

Apache Kafka is a distributed data store optimized for ingesting and processing stream data in real-time. Streaming data is data that is continuously generated by thousands of data sources which typically send the data records in simultaneously. It is a data pipeline reliably processes and moves data from one system to another, and a streaming application is an application that consumes streams of data. For example, if you want to create a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline. Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications [33].

3.4. Hive

Hive is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale. A data warehouse provides a central store of information that can easily be analyzed to make informed, data driven decisions. Hive allows users to read, write, and manage petabytes of data using SQL. Hive is built on top of Apache Hadoop, which is an open-source framework used to efficiently store and process large datasets [33].

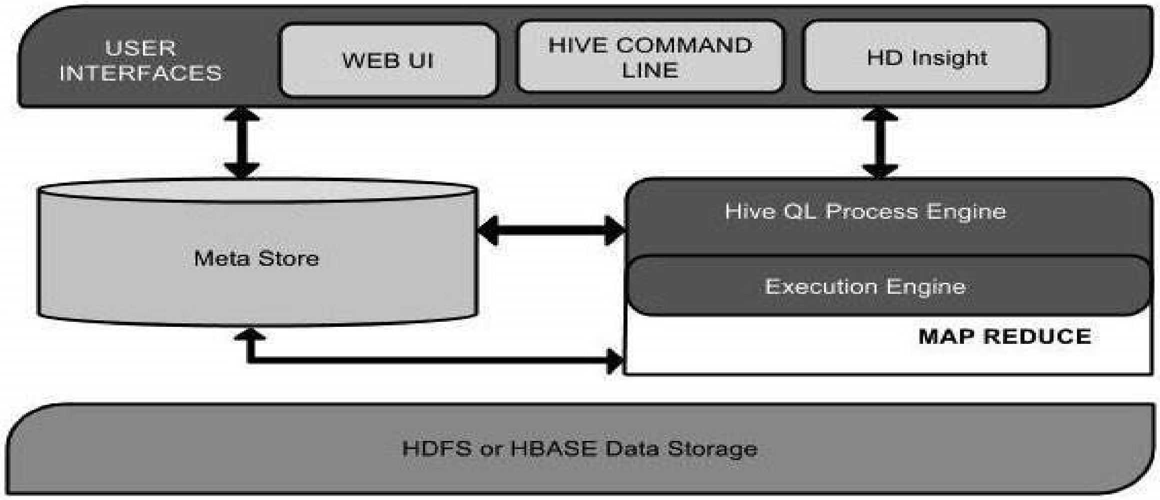


Fig 4: Hbase architecture [3]

3.5. Hbase

Hbase is a distributed column-oriented database built on top of the Hadoop file system (Fig. 4). It is an open-source project and is horizontally scalable. HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS). It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System. One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the

Hadoop File System and provides read and write access [34]. Fig. 4 shows the Hbase Architecture.

3.6. Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System (HDFS) is the primary data storage system used by Hadoop applications. As shown in Fig. 5, it employs a NameNode and DataNode architecture to implement a distributed file system that provides high-performance access to data across highly scalable Hadoop clusters. HDFS is a key part of the many Hadoop ecosystem technologies, as it provides a reliable means for managing pools of big data and supporting related big data analytics applications [34].

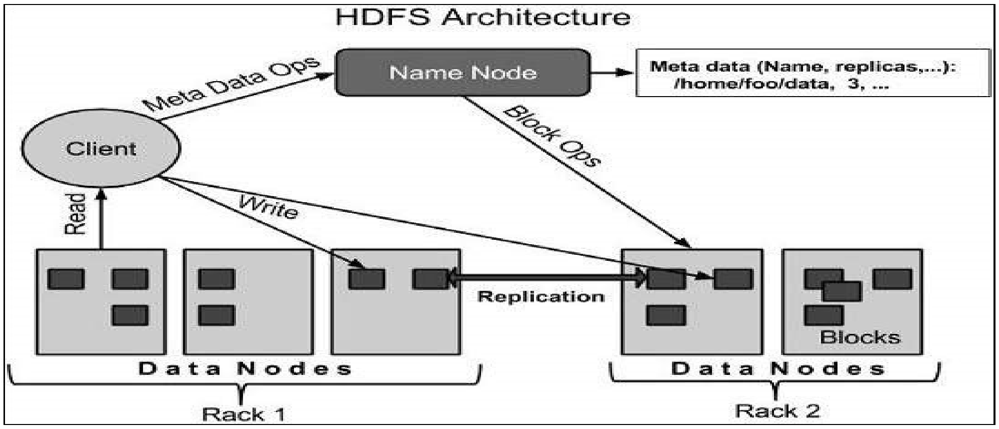


Fig 5: HDFS architecture [2]

3.7. Apache Spark

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.

Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming. Apart from supporting all these workloads in a respective system, it reduces the management burden of maintaining separate tools.

3.8. MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

3.9. Pig

Pig is a high-level programming language useful for analyzing large data sets. It is an abstraction over MapReduce. Pig is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop for data manipulation operations in Hadoop.

To write data analysis programs, Pig provides a high-level language known as Pig Latin. This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data [35].

3.10. Yarn

Yarn stands for "*Yet Another Resource Negotiator*". It was introduced in Hadoop 2.0 to remove the bottleneck on Job Tracker which was present in Hadoop 1.0. YARN was described as a "Redesigned Resource Manager" at the time of its launching, but it has now evolved to be known as large-scale distributed operating system used for Big Data processing [36].

3.11. Zookeeper

Zookeeper is a distributed co-ordination service to manage large set of hosts. Co-ordinating and managing a service in a distributed environment is a complicated process. ZooKeeper solves this issue with its simple architecture and API. ZooKeeper allows developers to focus on core application logic without worrying about the distributed nature of the application [36].

The ZooKeeper framework was originally built at "Yahoo!" for accessing their applications in an easy and robust manner. Later, Apache ZooKeeper became a standard for organized service used by Hadoop, HBase, and other distributed frameworks. For example, Apache HBase uses ZooKeeper to track the status of distributed data [36].

3.12. Apache Oozie

Apache Oozie is a scheduler system used to run and manage Hadoop jobs in a distributed environment. Oozie supports combining multiple complex jobs that run in a particular order for accomplishing a more significant task. With Oozie, within a particular set of tasks, two or more jobs can be programmed to run in parallel [36].

4. TRANSFORMATION TO NOSQL FROM RDMS

Now more data than ever before is being created, distributed and harnessed to make business decisions. In 2013, IBM said that 90% of the world's data had been created in the last 2 years alone [37].

The boom in unstructured data that the world has seen in the last few years is one of the main reasons relational databases are no longer sufficient for many companies' needs. One of reason we are seeing this boom in unstructured data is because of the global e access to the Internet. Contributing to this boom is the ubiquity of social media, wherein everybody wants to let others know happenings related to them as and when they are taking place. As more than 1/5th of the population is following such behavioral patterns, we can see that not only will data storage and fetching requirements become hugely important but simultaneously this also requires increased storage for various types of data like audio, video, images and textual data [37].

4.1. Relational Database

Data was originally stored in documents. However, as the quantity of information increased, accessing the information using files was not easy. It was a method that was slow and inefficient. As the quantity of information grew, keeping the information and collecting any records was very hard. Hierarchical and network databases were intended as mechanisms for storage, but they did not provide a normal technique for data access. SQL came into being with the need to handle information and the desire for a normal technique of accessing information [37].

4.2. ACID Properties

When a transaction system makes any transaction then the system has to ensure that transaction will meet a certain characteristic. Following are some properties that must be fulfilled when a transaction made:

- Atomicity: Every transaction is atomic mean to say if one part of the system fails the entire system fails.
- Consistency: Every transaction is subject to a set of rules.
- Isolation: No transaction interferes to another transaction.
- Durability: If any person is committed the transaction then other person gets the same committed data [37].

4.3. NoSql DataBase

As a technological environment transforms and faces new difficulties, companies progressively recognize that new methods and databases need to be evaluated to handle their information to help changing company needs and increasing complexity and development.

The Relational Database (RDBMS) was the dominant model for database administration. But non-relational, cloud or "NoSQL" databases are now emerging in common as an alternative model for database management [37].

The primary motive behind this strategy is: simpler design, simpler "horizontal" scaling to machine clusters, which is an issue for relational databases, and better accessibility control. The information structures used in NoSQL databases (e.g. key-value, graph, or document) are slightly different from those used in relational databases by default, making some activities in NoSQL quicker. The information structures used in NoSQL databases are also sometimes regarded as "more flexible" than in relational database tables. However, their total capabilities are still not disclosed [37].

In Big Data and real-time web applications, NoSQL databases are increasingly being used [37]. To emphasize that they can support SQLlike query languages, NoSQL systems are also sometimes called "Not only SQL".

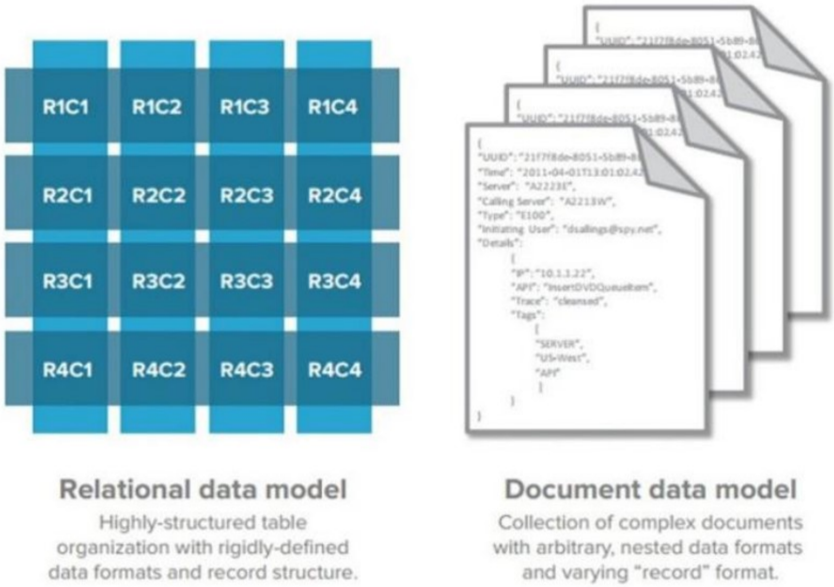
4.4. NoSQL Database Types

Many NoSQL databases are accessible, but they fall within four data models outlined in [3,12, 37]. Each category has its own particular characteristics, but the distinct information models are cross-checked. All NoSQL databases are generally designed for distribution and horizontal scaling, does not expose a SQL interface and may be open source. NoSQL databases vary depending on their data model in their performance [37].

4.4.1. Document Store Database

Document stored to databases in which information is stored in the form of documents. Document stores deliver excellent efficiency and choices for horizontal scalability. Documents within a document-oriented database are somewhat comparable to documents in relational databases, but are much more flexible because they are less schematic. The documents standard formats are like XML, PDF, JSON, and so on [37]. In relational databases, a record within the same database will have the same data fields and the unused data fields will be kept empty, but each document may have similar and dissimilar data in the case of document stores. A unique key that represents the document is used to address documents in the database. These keys can be a simple string or a URI or path string. In comparison with key value stores, document store is a little more complex because they allow the key value pairs to be embedded in documents which are also known as key document pairs.

For content management systems and blog applications, Document-oriented databases are suitable. Examples are the 10 G MongoDB, Apache CouchDB, Azure's DocumentDB and AWS DynamoDB, providers who use document orientated databases. The MongoDB is developed with a 10 G C++ and is an inter-plate based, cross-platform document-oriented database. Grid File System is used to store large files in binary JSON format such as images and videos. It delivers high efficiency, consistency and persistence but is not very reliable and has a hungry resource. Fig 6 shows the Document store in NoSql database architecture [37].



```
{
"UUID":"21f7f8de-8051-5b89-8
"Time": "2011-04-01T13:01:02.42
"Server": "A2223E",
"Calling Server": "A2213W",
"Type": "E100",
"Initiating User": "dsallings@spy.net",
"Details":
    {
    "IP":"10.1.1.22",
    "API": "InsertDVDQueueItem",
    "Trace": "cleansed",
    "Tags":
        [
        "SERVER",
        "US-West",
        "API"
        ]
    }
}
```

Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.

Document data model

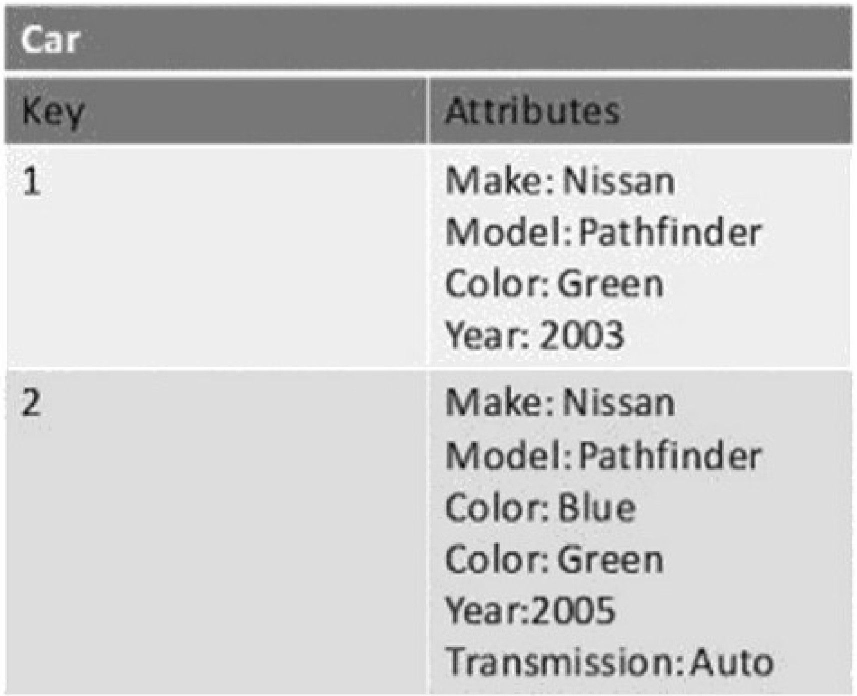
Collection of complex documents with arbitrary, nested data formats and varying "record" format.

Fig 6: Document store NoSQL database [30]

4.4.2. Key Value Store Database

The data stores with key-value are very simple, but they are silently effective and strong. The application program interface (API) is easy to use. The user can save the data in a schema less manner using key-value data store. The data is generally a type of programming language or object type of data. The information consists of 2 components, a string which depicts the key and the real value, producing a couple of "main value." The data saves are like hash tables in which keys are used for indexing, making them faster than RDBMS. The data model is therefore simple: a map and a dictionary that allows the user to request values based on specified key values. In modern data stores, information scalability is preferable to consistency. Therefore, ad-hoc querying and analytical characteristics such as links and aggregates were overlooked [38]. Key-value stores provide high competitiveness, quickly searching and mass storage choices. One of the weaknesses of key data store is the absence of a scheme to create a customized view of data.

Such key-value databases may be used as online shopping carts to create forums and websites for storing customer sessions. Amazon's DynamoDB, Cassandra, Azure Table Storage (ATS) are some remarkable examples. For internet scale apps Amazon provides DynamoDB's fully controlled NoSQL Store Service [39]. It is a distributed key value storage facility which, with its replica function, offers quick, safe, economical access to information and high availability and durability. Fig. 7 Key value store NoSQL database architecture.



Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Fig 7: Key value store NoSQL database [29]

4.4.3. Graph Stores Database

Graphs database are databases that store information as graphs as shown in Fig. 9. The graph contains nodes and edges, which maintain the relationships between the nodes and the items. The graph also includes node-related characteristics. It utilizes an index-free adjacency method that means that each node comprises of a direct point that points towards the neighboring node. This method allows millions of documents to be accessed. The primary focus on the association between information, in a graph database [39]. Graph databases provide less effective schematic and semi-structured data storage. The queries are articulated as crossover, thus increasing the speed of graph databases over relation databases. It is simple to measure and simple to use whiteboards. Graph databases comply with ACID and promote rollback.

These data bases are designed for the development of social networking apps, bioinformatics, content management systems and cloud management services. Notable graph databases are Neo4j, Orient DB, Apache Giraph, and Titan. Fig. 8 Graph store NoSQL database architecture.

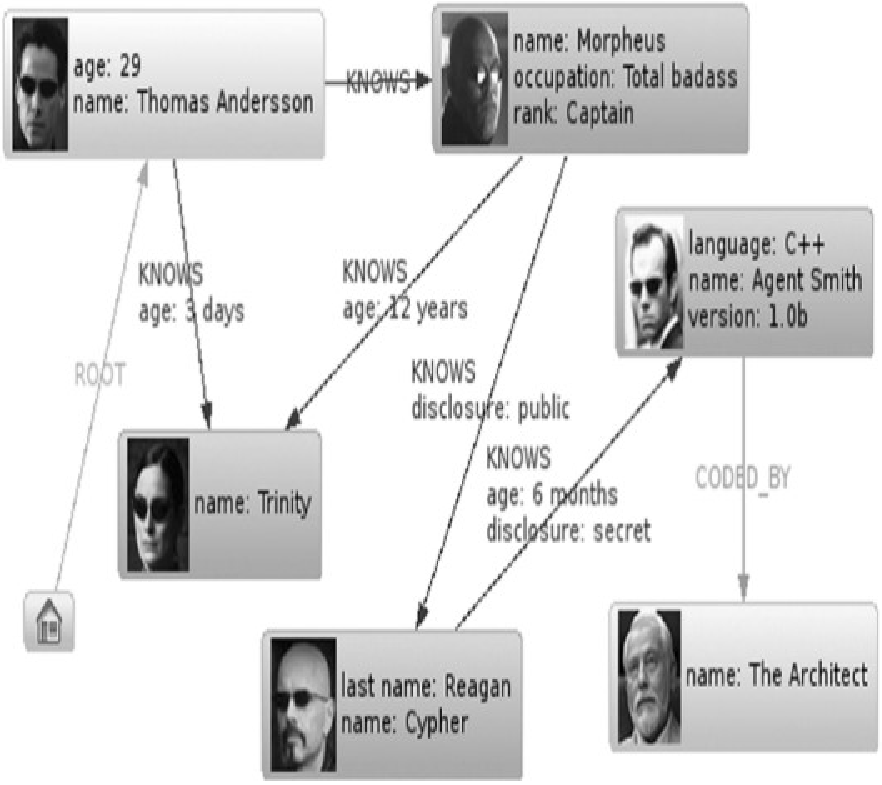
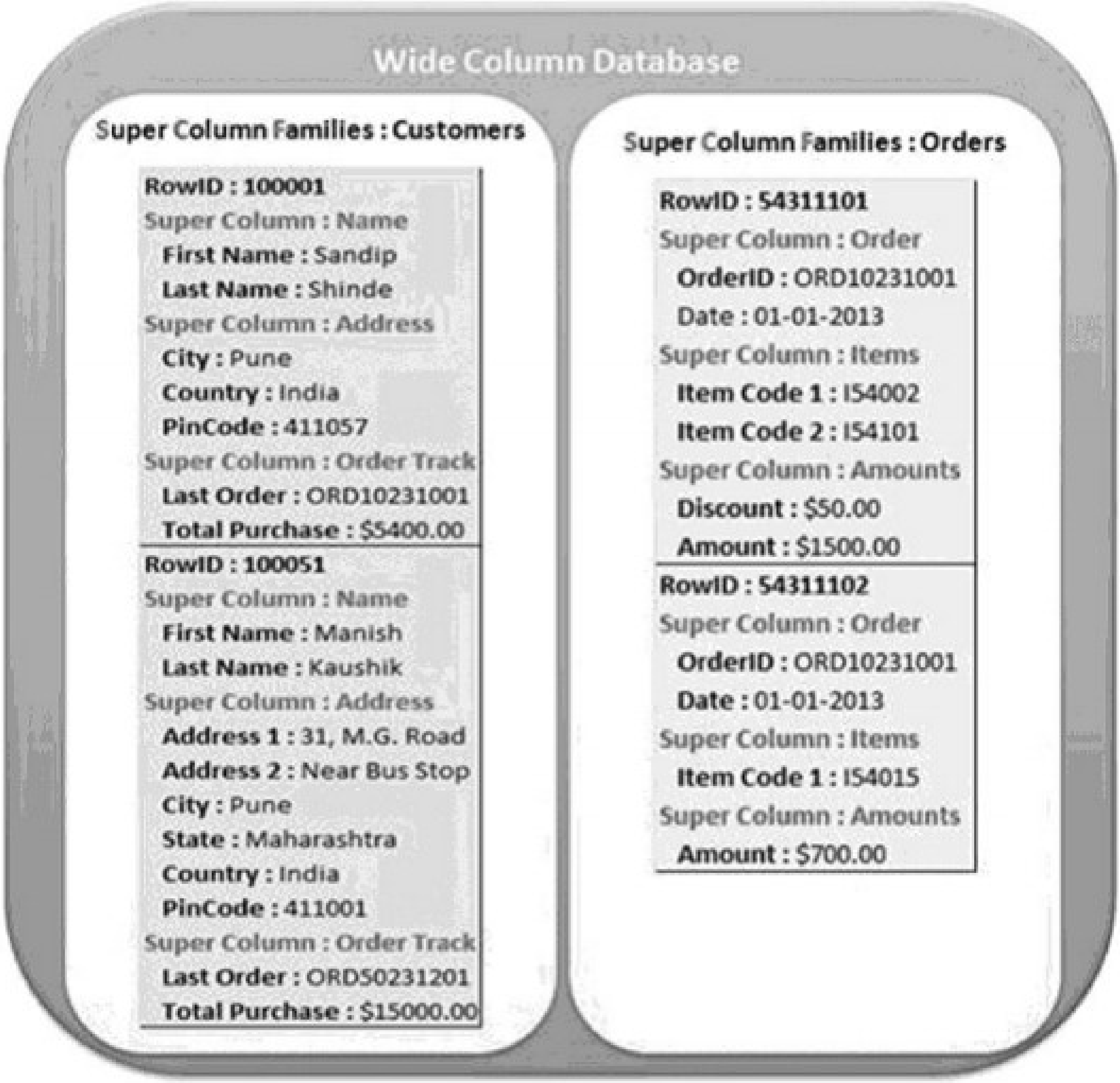


Fig 8: Graph store NoSQL database [12]

4.4.4. Wide column stores database

The NoSQL column stores are hybrid row / column stores as opposed to pure relational bases. Although column-by-column data storage and column additions to row-based databases are shared, column stores do not store database information in lists but store the information in massively distributed architectures. Each key has one or more characteristics (rows) for each row in column stores. A column store stores its information so that less I / O activity can be quickly added. It provides strong data storage scalability. The information saved in the database is based on the column family sort order.

Wide-column databases are perfect for data mining and Big Data analytics apps. Examples of column-oriented store suppliers include Cassandra (the high-performance of Facebook), Apache Hbase, Google's Big Table, and HyperTable. The Big Table by Google is a wide column high-performance database, able to handle large amounts of information. It has been created using C / C++ on Google File System GFS. It is used by several Google applications such as YouTube and Gmail that have different data base latency requirements [39]. Besides the use in the Google App Engine, it is not distributed outside of Google. Big Table is conceived for simple scalability on thousands of computers, so it is hardware-tolerant. Fig.b9. Wide Column Store NoSQL Database architecture.



Super Column Families : Customers

RowID : 100001
Super Column : Name
First Name : Sandip
Last Name : Shinde
Super Column : Address
City : Pune
Country : India
PinCode : 411057
Super Column : Order Track
Last Order : ORD10231001
Total Purchase : $5400.00

RowID : 100051
Super Column : Name
First Name : Manish
Last Name : Kaushik
Super Column : Address
Address 1 : 31, M.G. Road
Address 2 : Near Bus Stop
City : Pune
State : Maharashtra
Country : India
PinCode : 411001
Super Column : Order Track
Last Order : ORD50231201
Total Purchase : $15000.00

Super Column Families : Orders

RowID : 54311101
Super Column : Order
OrderID : ORD10231001
Date : 01-01-2013
Super Column : Items
Item Code 1 : I54002
Item Code 2 : I54101
Super Column : Amounts
Discount : $50.00
Amount : $1500.00

RowID : 54311102
Super Column : Order
OrderID : ORD10231001
Date : 01-01-2013
Super Column : Items
Item Code 1 : I54015
Super Column : Amounts
Amount : $700.00

Fig 9: Wide column store NoSQL database [29]

4.4.5. Comparison between RDMS and NoSql DataBase

The main reason why one would need to move to NoSQL databases is necessity in huge data storage (also called Big Data), scalability and performance reasons. Here are some tables displaying the difference in terminology and data operations between a NoSQL database and RDBMS (SQL).

We have made a high-level comparison between the SQL (relational) and NoSQL (non-relational) databases on the basis of the characteristics of each database type lately reported in the literature [40].

Table 1: Summarized view of SQL and NoSQL

SQL	MONGODB
Based on ACID transactional properties such as atomicity, consistency, isolation.	Supports AID transactions and CAP theorem of distributed systems support consistency of data across all nodes of a NoSQL database.
It has vertical Scaling.	It has horizontal Scaling.
Structured Query Language are used to manipulate the data.	Query the Data efficiently. Object oriented APIs are used.
Based on pre-defined foreign keys relationships between tables in an explicit database schema. Strict definition of schemas and data type is required before inserting the data	Dynamic database schema. Do not force schema definition in advance. Different data can be store together as required.
Softwares that use for this DB are oracle, MySQL, SQL Server.	MongoDB, Riak, Couchbase, Cassandra.

NoSQL Benefits over RDMS:

- Provides a wide selection of data models
- Easily scalable
- Administrators of the database are not necessary
- Some NoSQL DB suppliers such as Riak and Cassandra can manage hardware failure
- Faster, more efficient and flexible

- Has developed very rapidly
- Used for Big data applications

Though ACID makes a strong to relational databases some drawback of type is, these databases were works with single server that means way of enhancing the capacity is only upgrading the server. Another researcher, Antro Salminen, note that in his seminar that for scale–up the RDBMS only way is by adding the Hardware processing power [12] RDBMS also has physical storage limit [3]. Hence such database was required that support current generation web, mobile, and other applications to operate at any scale. As need is a Mother of innovations, when everyone was facing a problem of Big data there was an introduction of NoSQL by the researchers.

NoSQL gives the benefit of time over relational databases' join queries by providing a feature of simple graph traversal operation queries. We can store frequently required information in one table in NoSQL with its horizontal Scaling property. NoSQL also able to handle all the table Joins at application level. Another feature of NoSQL is Data repetition is acceptable. This feature of NoSQL helps to improve efficiency and execution speed of query. However, tables are not related in NoSQL so we need to very careful and needs to synchronize the data while updating the Tables [40].

4.4.5. Performance of NoSql and Sql Database for Big Data

The main reason to move to NoSQL from the relational database is because of performance improvement demands. Choi et al. discovered that the database NoSQL such as MongoDB offered quicker and more stable results at the cost of information coherence. Testing was performed on the basis of an open source project on an internal blog system MongoDB has found that 85% faster than a SQL database have stored posts. NoSQL was proposed in settings that relate to information accessibility rather than consistency.

The usage of MongoDB in mobile apps is described by Fotache & Cogean [40]. Some various updating operations, such as Upsert, are simpler and quicker than the SQL database with NoSQL. The use of cloud computing and NoSQL will improve the performance of mobile platforms, especially in the data layer.

In the case of a Triple Store based on Resource Description Framework (RDF) as a NoSQL database, the Ullah [40] has contrasted the results of both the relational database management scheme (RDBMS) and NoSQL. Reading a great deal of data is very comprehensive in the database and because the NoSQL database is un-structured, the storage of thousands of records is a huge amount while RDBMS uses less storage [40]. For example, the NoSQL database search for white hat took 5255 ms and only 165.43 ms to the RDBMS.

The Yahoo Cloud Serving Benchmark (YCSB) experiment was carried out by Floratou etc. [40] on RDBMS and MongoDB. They have tested the SQL client shared database for MongoDB and client shared databases. Tests showed that most of the benchmarks achieved higher output and reduced latency with SQL client-shared databases. The reason for higher performance is that most read requests have been sent to pages in the buffer pool whereas the NoSQL databases tend to read shards on different nodes [40]. The research has proven that the processing power for RDBMS remains the same for NoSQL to deal with greater workloads.

5. DESCRIPTION OF DATASETS

Since this paper is about analyzing tools and making a comparative study between them, so I needed datasets that are unique from each other and large enough to work with. There are many online sources where from we can collect data. So, I collected data from two of the popular data sources which are US DATA.GOV (https://www.data.gov/), Kaggle (https://www.kaggle.com/).

Below, I tried to provide a small description of some of the datasets that were used:

5.1. Iris Data Set

This is perhaps the best-known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

- Predicted attribute: class of iris plant.

This is an exceedingly simple domain (source: https://archive.ics.uci.edu/ml/datasets/Iris).

5.2. Wine Data Set

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines (Source: https://archive.ics.uci.edu/ml/datasets/Wine).

5.3. Liver Disorders Data Set

The first 5 variables are all blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Each line in the dataset constitutes the record of a single male individual.

Important note: The 7th field (selector) has been widely misinterpreted in the past as a dependent variable representing presence or absence of a liver disorder. This is incorrect [40]. The 7th field was created by BUPA researchers as a train/test selector. It is not suitable as a dependent variable for classification. The dataset does not contain any variable representing presence or absence of a liver disorder. Researchers who wish to use this dataset as a classification benchmark should follow the method used in experiments by the donor (Forsyth & Rada, 1986, Machine learning: applications in expert systems and information retrieval) and others (e.g. Turney, 1995, Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm), who used the 6th field (drinks), after dichotomising, as a dependent variable for classification. Because of widespread misinterpretation in the past, researchers should take care to state their method clearly (Source: https://archive.ics.uci.edu/ml/datasets/liver+disorders).

5.4. Heart Disease Data Set

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

The names and social security numbers of the patients were recently removed from the database, replaced with dummy values. One file has been "processed", that one containing the Cleveland database. All four unprocessed files also exist in this directory (source: https://archive.ics.uci.edu/ml/datasets/Heart+Disease)

5.5. Solar power Generated Data

This data has been gathered at two solar power plants in India over a 34 day period. It has two pairs of files - each pair has one power generation dataset and one sensor readings dataset. The power generation datasets are gathered at the inverter level - each inverter has multiple lines of solar panels attached to it. The sensor data is gathered at a plant level - single array of sensors optimally placed at the plant (source: https://www.kaggle.com/anikannal/solar-power-generation-dataEE)

6. EXPLANATION OF THE EXPERIMENT

My experiment involves few steps which are mentioned below:

6.1. Gathering data

Collecting data was the first step involved in this experiment. I went through various sources namely DATA.GOV (www.data.gov), Amazon public data sets (registry.opendata.aws), reddit (www.reddit.com/r/datasets/), Kaggle (kaggle.com) in order to gather the right dataset.

6.2. Cleaning

Once the data is collected, the next step is to clean it. Its critical cause in order to analyze the data, it needs to be in right order. There are lots of tools available these days to clean the dataset but the one I preferred was Microsoft Power BI. So, power BI tool was used to clean the datasets.

6.3. Language

I used python as programming language. Python programming involves fewer lines of codes as compared to other languages available for programming. It is able to execute programs in the least lines of code. Moreover, Python automatically offers assistance to identify and associate data types.

Python programming follows an indentation-based nesting structure. The language can process lengthy tasks within a short span of time. As there is no limitation to data processing, you can compute data in commodity machines, laptop, cloud, and desktop.

- Earlier, Python was considered to be a slower language in comparison to some of its counterparts like Java and Scala but the scenario has changed now. The advent of *t*he Anaconda platform has offered a great speed to the language. This is why Python for big data has become one of the most popular options in the industry. You can also hire Python Developer who can implement these Python benefits in your business.

6.4. Platform

For the better performance of the research, I needed cluster computer. A computer cluster is a set of connected computers (nodes) that work together as if they are a single (much more powerful) machine. Unlike grid computers, where each node performs a different task, computer clusters assign the same task to each node. Nodes in a cluster are usually connected to each other through high-speed local area networks. Each node runs its own instance of an operating system. A computer cluster may range from a simple two-node system connecting two personal computers to a supercomputer with a cluster architecture. Computer clusters are often used for cost-effective high performance computing (HPC) and high availability (HA) by businesses of all sizes. If a single component fails in a computer cluster, the other nodes continue to provide uninterrupted processing. Compared to a single computer, a computer cluster can provide faster processing speed, larger storage capacity, better data integrity, greater reliability and wider availability of resources. Computer clusters are usually dedicated to specific functions, such as load balancing, high availability, high performance or large-scale processing. Compared to a mainframe computer, the amount of power and processing speed produced by a cluster is more cost effective. The networked nodes in a cluster also create an efficient, distributed infrastructure that prevents bottlenecks, thus improving performance [41].

Hence, I used a lab named ITVersity Labs that provides the perfect environment to learn essential skills in various Big Data technologies. They have the cluster computer with 12 nodes (computer) that helped me to run the testing at a quite faster pace with greater accuracy. Moreover, their technical team was also supportive in taking care of the technical issues.

6.5. Framework

I used apache Hadoop as frame work. It is an open-source data platform or framework developed in Java, dedicated to store and analyze large sets of unstructured data [42].

6.6. Features of Apache Hadoop

- Allows multiple concurrent tasks to run from single to thousands of servers without any delay.
- Consists of a distributed file system that allows transferring data and files in split seconds between different nodes.

Able to process efficiently even if a node fails [42].

7. COMPARATIVE STUDY AMONG VARIOUS BIG DATA TOOLS

7.1. Tez vs MapReduce

Distributed processing is the base of Hadoop. Hive relies on MapReduce for distributed processing. However, MapReduce is batch oriented so, it is not suitable for interactive queries. But apache Tez is the alternative for interactive query processing. Tez is prominent over MapReduce by using Hadoop containers efficiently, multiple reduce phases without map phases and effective use of HDFS.

MapReduce always requires a map phase before the reduce phase. Stores temporary data into HDFS after every map and reduce phase. This is slower due to the access of HDFS after every Map and Reduce phase. A single Map phase may have multiple reduce phase. MapReduce programs are written in different programming and scripting languages. It is a framework which helps in writing programs for processing of data in parallel across thousands of machines. MapReduce is a framework which helps in writing programs for processing of data in parallel across thousands of machines.

Tez represents the MapReduce paradigm in a more powerful framework based on expressing computations as a dataflow graph. Tez enables developers to build end-user applications with better performance and flexibility. Hadoop is a batch-processing platform for large amounts of data. However, there are a lot of use cases for near-real-time performance of query processing. There are also several workloads, such as Machine Learning, which do not fit well into the MapReduce paradigm. Tez helps Hadoop to address these use cases.

Tez is a highly customizable framework that meets broad spectrum of users. Moreover, projects such as Hive and Pig are seeing significant improvements in response times when they use Tez instead of MapReduce as the backbone for data processing. Tez is built on top of YARN, which is also the new resource-management framework for Hadoop [43].

Fig. 10 and 11 show the Hive and Tez Architecture, respectively.

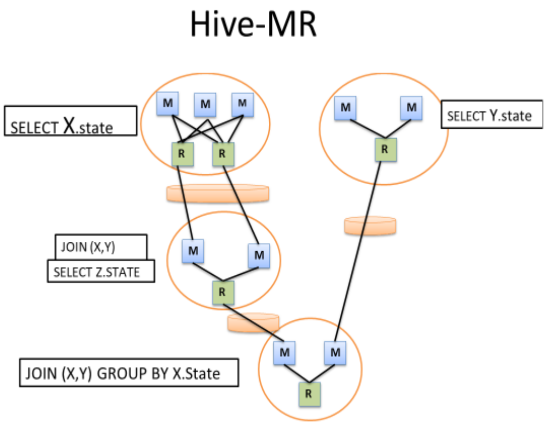


Fig 10: Hive- MR architecture [1]

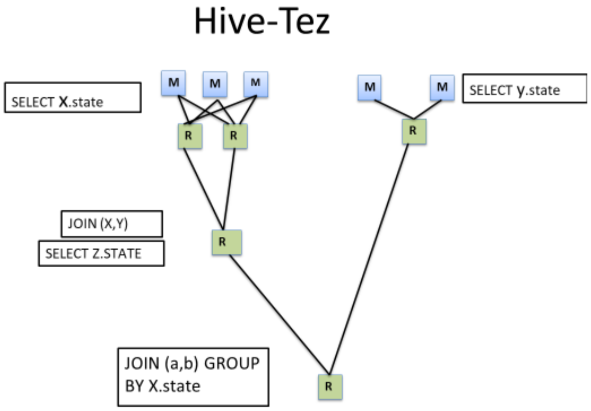


Fig 11: Hive-Tez architecture [1]

7.1.1. Experiment

In order to compare the performance between MapReduce and Tez, I tried to read 15 datasets using python programming and applied mapReduce and Tez on them.

7.1.2. MapReduce Code

- Map:

```
import sys
for line in sys.stdin:
           line = line.strip()
           words = line.split()
           for word in words:
                    print '%s\t%s' % (word, 1)
```

- Reduce:

```
from operator import itemgetter
import sys
word2count = {}
for line in sys.stdin:
            line = line.strip()
             word, count = line.split('\t', 1)
            try:
                  count=int(count)
                  word2count[word]=word2count.get(word,0)+count
            except ValueError:
                        pass
sorted_word2count = sorted(word2count.items(), key= itemgetter(0))
for word, count in sorted_word2count:
              print '%s\t%s'% (word, count)
```

- Tez Code:

```
@InterfaceAudience.Private
    public abstract class TezExampleBase extends Configured implements Tool {

      private static final Logger LOG = LoggerFactory.getLogger(TezExampleBase.class);

      private TezClient tezClientInternal;
      protected static final String DISABLE_SPLIT_GROUPING = "disableSplitGrouping";
      protected static final String LOCAL_MODE = "local";
      protected static final String COUNTER_LOG = "counter";
      protected static final String GENERATE_SPLIT_IN_CLIENT = "generateSplitInClient";
      protected static final String LEAVE_AM_RUNNING = "leaveAmRunning";
      protected static final String RECONNECT_APP_ID = "reconnectAppId";
      private boolean disableSplitGrouping = false;
      private boolean isLocalMode = false;
      private boolean isCountersLog = false;
      private boolean generateSplitInClient = false;
```

```
  private boolean leaveAmRunning = false;
  private String reconnectAppId;
  private HadoopShim hadoopShim;

  protected boolean isCountersLog() {
         return isCountersLog;
  }

  protected boolean isDisableSplitGrouping() {
    return disableSplitGrouping;
  }

  protected boolean isGenerateSplitInClient() {
    return generateSplitInClient;
  }

  private Options getExtraOptions() {
    Options options = new Options();
    options.addOption(LOCAL_MODE, false, "run it as local mode");
    options.addOption(DISABLE_SPLIT_GROUPING, false , "disable split grouping");
    options.addOption(COUNTER_LOG, false , "print counter log");
    options.addOption(GENERATE_SPLIT_IN_CLIENT, false, "whether generate split in
client");
    options.addOption(LEAVE_AM_RUNNING, false, "whether client should stop session");
    options.addOption(RECONNECT_APP_ID, true, "appId for client reconnect");
    return options;
  }

  @Override
  public final int run(String[] args) throws Exception {
    Configuration conf = getConf();
    GenericOptionsParser optionParser = new GenericOptionsParser(conf,
getExtraOptions(), args);
    String[] otherArgs = optionParser.getRemainingArgs();
    if (optionParser.getCommandLine().hasOption(LOCAL_MODE)) {
      isLocalMode = true;
    }
    if (optionParser.getCommandLine().hasOption(DISABLE_SPLIT_GROUPING)) {
      disableSplitGrouping = true;
    }
    if (optionParser.getCommandLine().hasOption(COUNTER_LOG)) {
      isCountersLog = true;
    }
    if (optionParser.getCommandLine().hasOption(GENERATE_SPLIT_IN_CLIENT)) {
      generateSplitInClient = true;
    }
    if (optionParser.getCommandLine().hasOption(LEAVE_AM_RUNNING)) {
      leaveAmRunning = true;
    }
    if (optionParser.getCommandLine().hasOption(RECONNECT_APP_ID)) {
        reconnectAppId =
optionParser.getCommandLine().getOptionValue(RECONNECT_APP_ID);
    }
    hadoopShim = new HadoopShimsLoader(conf).getHadoopShim();

    return _execute(otherArgs, null, null);
  }

  /**
   * Utility method to use the example from within code or a test.
   *
   * @param conf      the tez configuration instance which will be used to crate the
DAG and
   *                  possible the Tez Client.
   * @param args      arguments to the example
   * @param tezClient an existing running {@link org.apache.tez.client.TezClient}
instance if one
   *                  exists. If no TezClient is specified (null), one will be created
based on the
   *                  provided configuration. If TezClient is specified, local mode
option can not been
   *                  specified in arguments, it takes no effect.
   * @return Zero indicates success, non-zero indicates failure
   * @throws Exception
   */
  public int run(TezConfiguration conf, String[] args, @Nullable TezClient tezClient)
throws
      Exception {
    setConf(conf);
    hadoopShim = new HadoopShimsLoader(conf).getHadoopShim();
    GenericOptionsParser optionParser = new GenericOptionsParser(conf,
getExtraOptions(), args);
    if (optionParser.getCommandLine().hasOption(LOCAL_MODE)) {
      isLocalMode = true;
      if (tezClient != null) {
        throw new RuntimeException("can't specify local mode when TezClient is created,
it takes no effect");
      }
    }
    if (optionParser.getCommandLine().hasOption(DISABLE_SPLIT_GROUPING)) {
      disableSplitGrouping = true;
    }
    if (optionParser.getCommandLine().hasOption(COUNTER_LOG)) {
      isCountersLog = true;
    }
    if (optionParser.getCommandLine().hasOption(GENERATE_SPLIT_IN_CLIENT)) {
      generateSplitInClient = true;
    }
    String[] otherArgs = optionParser.getRemainingArgs();
    return _execute(otherArgs, conf, tezClient);
  }

  /**
   * @param dag           the dag to execute
   * @param printCounters whether to print counters or not
   * @param logger        the logger to use while printing diagnostics
   * @return Zero indicates success, non-zero indicates failure
   * @throws TezException
   * @throws InterruptedException
   * @throws IOException
   */
  public int runDag(DAG dag, boolean printCounters, Logger logger) throws TezException,
      InterruptedException, IOException {
    tezClientInternal.waitTillReady();

    CallerContext callerContext = CallerContext.create("TezExamples",
        "Tez Example DAG: " + dag.getName());
    ApplicationId appId = tezClientInternal.getAppMasterApplicationId();
    if (hadoopShim == null) {
      Configuration conf = (getConf() == null ? new Configuration(false) : getConf());
      hadoopShim = new HadoopShimsLoader(conf).getHadoopShim();
    }

    if (appId != null) {
      TezUtilsInternal.setHadoopCallerContext(hadoopShim, appId);
      callerContext.setCallerIdAndType(appId.toString(), "TezExampleApplication");
    }
    dag.setCallerContext(callerContext);

    DAGClient dagClient = tezClientInternal.submitDAG(dag);
    Set<StatusGetOpts> getOpts = Sets.newHashSet();
    if (printCounters) {
      getOpts.add(StatusGetOpts.GET_COUNTERS);
    }

    DAGStatus dagStatus;
    dagStatus = dagClient.waitForCompletionWithStatusUpdates(getOpts);

    if (dagStatus.getState() != DAGStatus.State.SUCCEEDED) {
      logger.info("DAG diagnostics: " + dagStatus.getDiagnostics());
```

```
      return -1;
    }
    return 0;
  }

  private int _validateArgs(String[] args) {
    int res = validateArgs(args);
    if (res != 0) {
      _printUsage();
      return res;
    }
    return 0;
  }

  private int _execute(String[] otherArgs, TezConfiguration tezConf, TezClient
tezClient) throws
      Exception {

    int result = _validateArgs(otherArgs);
    if (result != 0) {
      return result;
    }

    if (tezConf == null) {
      tezConf = new TezConfiguration(getConf());
    }
    if (isLocalMode) {
      LOG.info("Running in local mode...");
      tezConf.setBoolean(TezConfiguration.TEZ_LOCAL_MODE, true);
      tezConf.set("fs.defaultFS", "file:///");
      tezConf.setBoolean(
          TezRuntimeConfiguration.TEZ_RUNTIME_OPTIMIZE_LOCAL_FETCH, true);
    }
    UserGroupInformation.setConfiguration(tezConf);
    boolean ownTezClient = false;
    if (tezClient == null) {
      ownTezClient = true;
      tezClientInternal = createTezClient(tezConf);
    } else {
      tezClientInternal = tezClient;
    }
    try {
      return runJob(otherArgs, tezConf, tezClientInternal);
    } finally {
      if (ownTezClient && tezClientInternal != null && !leaveAmRunning) {
        tezClientInternal.stop();
      }
    }
  }

  private TezClient createTezClient(TezConfiguration tezConf) throws IOException,
TezException {
    TezClient tezClient = TezClient.create("TezExampleApplication", tezConf);
    if(reconnectAppId != null) {
      ApplicationId appId = TezClient.appIdfromString(reconnectAppId);
      tezClient.getClient(appId);
    } else {
      tezClient.start();
    }
    return tezClient;
  }

  private void _printUsage() {
    printUsage();
    System.err.println();
    printExtraOptionsUsage(System.err);
    System.err.println();
    ToolRunner.printGenericCommandUsage(System.err);
  }

  /**
   * Print usage instructions for this example
   */
  protected abstract void printUsage();

  protected void printExtraOptionsUsage(PrintStream ps) {
    ps.println("Tez example extra options supported are");
    ps.println("-" + LOCAL_MODE + "\t\trun it in tez local mode, "
        + " run it in distributed mode without this option");
    ps.println("-" + DISABLE_SPLIT_GROUPING + "\t\t disable split grouping for
MRInput,"
        + " enable split grouping without this option.");
    ps.println("-" + COUNTER_LOG + "\t\t to print counters information");
    ps.println("-" + GENERATE_SPLIT_IN_CLIENT + "\t\tgenerate input split in client");
    ps.println("-" + LEAVE_AM_RUNNING + "\t\twhether client should stop session");
    ps.println("-" + RECONNECT_APP_ID + "\t\tappId for client reconnect");
    ps.println();
    ps.println("The Tez example extra options usage syntax is ");
    ps.println("example_name [extra_options] [example_parameters]");
  }
  /**
   * Validate the arguments
   *
   * @param otherArgs arguments, if any
   * @return Zero indicates success, non-zero indicates failure
   */
  protected abstract int validateArgs(String[] otherArgs);

  /**
   * Create and execute the actual DAG for the example
   *
   * @param args      arguments for execution
   * @param tezConf   the tez configuration instance to be used while processing the
DAG
   * @param tezClient the tez client instance to use to run the DAG if any custom
monitoring is
   *                  required. Otherwise the utility method {@link
#runDag(org.apache.tez.dag.api.DAG,
   *                  boolean, org.slf4j.Logger)} should be used
   * @return Zero indicates success, non-zero indicates failure
   * @throws IOException
   * @throws TezException
   */
  protected abstract int runJob(String[] args, TezConfiguration tezConf,
                                TezClient tezClient) throws Exception;

  @Private
  @VisibleForTesting
  public ApplicationId getAppId() {
    if (tezClientInternal == null) {
      LOG.warn("TezClient is not initialized, return null for AppId");
      return null;
    }
    return tezClientInternal.getAppMasterApplicationId();
  }
}
```

Execution time taken by MR and Tez of the sample queries, which are listed below:

Table 2: Execution time between MR and Tez on sample queries

Query	MR (time in sec)	Tez (time in sec)
Q1	0.99	0.64
Q2	1.44	0.55
Q3	0.82	0.33
Q4	0.87	0.65
Q5	0.97	0.3
Q6	1.09	0.9
Q7	1.32	0.5
Q8	0.77	0.23
Q9	0.69	0.5
Q10	0.94	0.77
Q11	0.80	0.57
Q12	0.96	0.56
Q13	0.78	0.66
Q14	0.87	0.44
Q15	0.45	0.26

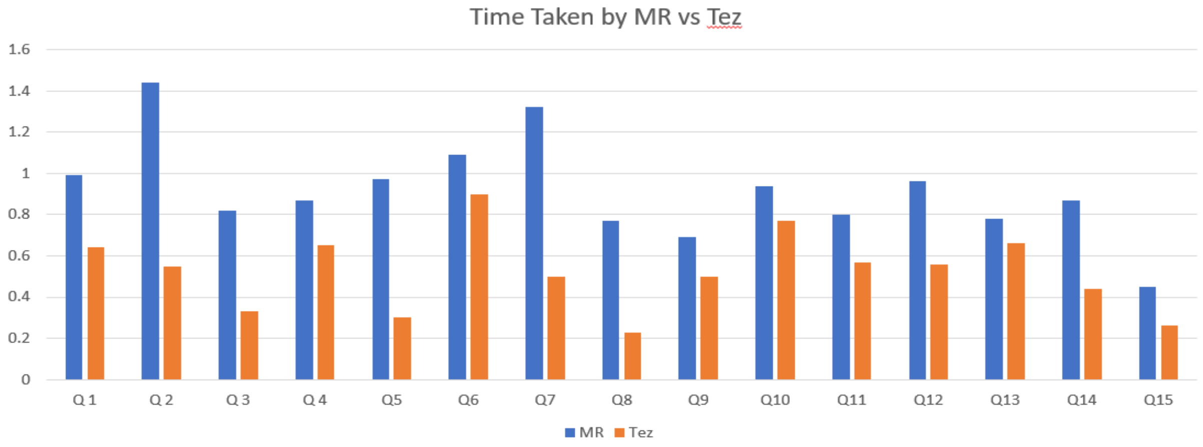


Fig 12: Time taken by MR vs Tez

7.2. Sqoop Vs Flume

Big Data is unquestionably synonymous with Apache Hadoop because of its cost-effectiveness and also for its virtues like scalability to process humongous loads of data. To get your data that needs to be analyzed on the Hadoop clusters is one of the most critical activities that can be done in any Big Data deployments. Data ingestion is the most critical activity as we just spoke about it, as it is required to load humongous loads of data in the orders of petabytes and exabytes.

Apache Sqoop and Apache Flume are two different technologies from the Hadoop ecosystem which can be put to use to gather data from various kinds of data sources and finally load that data into a traditional HDFS system. Apache Sqoop in Hadoop is used to fetch structured data from RDBMS systems like Teradata, Oracle, MySQL, MSSQL, PostgreSQL and on the other hand Apache Flume is used to fetch data that is stored on various sources as like the log files on a Web Server or an Application Server.

Apache Sqoop, which can be comfortably referred to as SQL to Hadoop is a lifesaver for any individual who experiences difficulties in moving data from data warehouses to the orthodox Hadoop environments. It is a very efficient and an effective Hadoop tool that can be used to import data from the traditional RDBMS onto HBase, Hive or HDFS. Apache Sqoop can also be used for the reverse use cases as well, that is to import data from a traditional HDFS to an orthodox RDBMS system too.

Apache Sqoop is an effective Hadoop related tool for all non-programmers to look at the RDBMS that needs to be imported into HDFS systems. Once the input is identified by Apache Sqoop, metadata on the table can be read and a specific class definition is created for the input requirements. Apache Sqoop can also be brutally forced to obtain the details of columns that are required before input instead of importing the whole input and saves a great amount of time in the process of it [43].

The most important features of Apache Flume are provided as below, let us now take a look at the following features:

- Apache Sqoop supports bulk import
- Sqoop allows parallel data transfers for optimal utilization of system resources and also to ensure faster performances
- Sqoop is made to increase the data analysis efficiency by a great deal
- Sqoop helps in mitigating excessive loads on external systems
- Sqoop provides interaction with the data programmatically by generating Java classes

Apache Flume can be explained as a service that is designed specifically to stream logs into Hadoop's environment. Apache Flume is a distributed and a reliable source to collect, aggregate larger amounts of log data. Apache Flume's architecture is specifically based on streaming data flows which is quite simple and makes it easier to use. Apache Flume provides many tunable reliability mechanisms, recovery and failover mechanisms that come to our rescue at the right time.

Apache Flume has a very simple event-driven approach with very important roles like Source, Channel and Sink.

- A Source is defined as the point from where the data comes (e.g., Message queue or a file)
- A Sink is defined as the point of data pipelined from various sources
- A Channel is defined as the pipes that establish connections between Sources and Sinks

Apache Flume works on two major concepts as discussed below:

- Master acts as a reliable configuration service that is used by nodes to retrieve their specific configurations
- Change in the configuration for a particular node on the Master is dynamically updated by the Master itself.

A node is generally an event pipe in Apache Hadoop Flume that reads from a Source and writes to a Sink. The characteristics and the roles of an Apache Flume node can be determined by the behavior of Sources and Sinks. Apache Flume was developed in such a manner as if the various options of Sources and Sinks do not match the requirements, then custom Sources and Sinks can be written to answer the needs.

The most important features of Apache Flume are provided as below, let us now take a look at the following features:

- Apache Flume is a flexible tool that enables scalability in the environments
- Flume provides very high throughput and at a very low latency
- Flume has a nice way of declarative configuration and alongside with it the ease of extensibility
- Flume in Hadoop is known to be fault tolerant, linearly scalable and also stream-oriented

Big Data systems, in general, are very popular and are known to be able to process huge amounts of unstructured and structured data from various kinds of data sources. The complexity of big data system increases with the data sources available. With diverse data sources and data from these data sources can be consistently produced on a large scale.

Apache Sqoop is a lifesaver in moving data from the data warehouse into the Hadoop environment. Interestingly it named Sqoop as SQL-to-Hadoop. Basically, for importing data from RDBMS's like MySQL, Oracle, etc. into HBase**,** Hive or HDFS**.** Apache Sqoop is an effective Hadoop tool. Also, user can export data from HDFS into RDBMS through Sqoop. In addition, Sqoop is a command line interpreter. Since interpreter executes Sqoop commands one at a time.

To be specific Sqoop is used for parallel data transfer. For this reason, the output could be in multiple files. It has a connector-based architecture. So, connectors know how to connect to the respective data source and fetch the data. HDFS is designated for data import using sqoop and it is not event driven. In order to import data from structured data sources, one has to use Sqoop only, because its connectors know how to interact with structured data sources and fetch data from them. Sqoop primarily used for copying data faster and then using it for generating analytical outcomes. Sqoop reduces the excessive storage and processing loads by transferring them to other system and has fast performance [43].

Sample sqoop command to import data from warehouse and target directory:

- Sqoop import from data warehouse:

```
--connect jdbc: mysql://ms.itversity.com:3306/retail_db \
--username retail_user \
-- password ******* \
-- table order_items \
-- warehouse-dir /user/dgadiraju/sqoop_import/retail_db
```

- Sqoop import from data target directory:

```
--connect jdbc: mysql://ms.itversity.com:3306/retail_db \
--username retail_user \
-- password ******* \
-- table order_items \
-- target-dir /user/dgadiraju/sqoop_import/retail_db/order_items
```

Apache flume is used for streaming logs into Hadoop environment, Apache Flume is best service designed for collecting and aggregating huge amounts of log data, jms, directory, crash reports etc. Flume is a distributed and reliable service. Moreover, it has very simple and easy to use architecture, on the basis of streaming data flows. Flume has an agent-based architecture. Here, a code is written (which is called as 'agent') which takes care of fetching data. Also, it has tunable reliability mechanisms and several recoveries and failover mechanisms. In flume, data flows to hdfs through zero to more channels. Here the data load can be driven by an event.

Generally, flume is used to pull data when companies want to analysis patterns, root causes or sentiment analysis using logs and social media. Lastly, is fault tolerant, robust and has tenable reliability mechanism for failover and recovery.

Sample code for importing data from log server*:*

```
##wshdfs.conf
  # To get the data from web server logs to HDFS
  wh.sources = ws
  wh.sinks = hd
  wh.channels = mem

  # Describe/configure the source
  wh.sources.ws.type = exec
  wh.sources.ws.command = tail -F /opt/gen_logs/logs/access.log

  # Describe the sink
  wh.sinks.hd.type = hdfs
  wh.sinks.hd.hdfs.path = hdfs://nn01.itversity.com:8020/user/dgadiraju/flume_demo

  wh.sinks.hd.hdfs.filePrefix = FlumeDemo
  wh.sinks.hd.hdfs.fileSuffix = .txt
  wh.sinks.hd.hdfs.rollInterval = 120
  wh.sinks.hd.hdfs.rollSize = 1048576
  wh.sinks.hd.hdfs.rollCount = 100
  wh.sinks.hd.hdfs.fileType = DataStream

  # Use a channel which buffers events in memory
  wh.channels.mem.type = memory
  wh.channels.mem.capacity = 1000
  wh.channels.mem.transactionCapacity = 100

  # Bind the source and sink to the channel
  wh.sources.ws.channels = mem
  wh.sinks.hd.channel = mem
```

7.3. Why Spark is Faster Than MapReduce

Big data analytics is one of the most active research areas with a lot of challenges and needs for new innovations that affect a wide range of industries. To fulfill the computational requirements of massive data analysis, an efficient framework is essential to design, implement and manage the required pipelines and algorithms. In this regard, Apache Spark has emerged as a unified engine for large-scale data analysis across a variety of workloads. It has introduced a new approach for data science and engineering where a wide range of data problems can be solved using a single processing engine with general-purpose languages. Following its advanced programming model, Apache Spark has been adopted as a fast and scalable framework in both academia and industry. It has become the most active big data open source project and one of the most active projects in the Apache Software Foundation.

As an evolving project in the big data community, having good references is a key need to get most of the Apache Spark and contribute effectively to its progress. While the official programming guide is the most up-to-date source about Apache Spark, several books have been published to show how Apache Spark can be used to solve big data problems. In addition, Databricks, the company founded by the creators of Apache Spark, has developed a set of reference applications to demonstrate how Apache Spark can be used for different workloads. Other good sources are the official blog at Databricks and Spark Hub where you can find Spark's news, events, resources, etc. However, the rapid adoption and development of Apache Spark, coupled with an increasing research on using it for big data analytics, make it difficult for beginners to comprehend the full body of development and research behind it. To our knowledge, there is no comprehensive summary on big data analytics using Apache Spark.

In order to fill this gap, help in getting started with Apache Spark and follow such an active project, the goal of this paper is to provide a concise succinct source of information about the key features of Apache Spark. Specifically, we focus on how Apache Spark can enable efficient large-scale machine learning, graph analysis and stream processing.

Apache Spark processes data in random access memory (RAM), while Hadoop MapReduce persists data back to the disk after a map or reduce action. In theory, then Spark should outperform Hadoop MapReduce. Spark can do more than plain data processing: it can also process graphs, and it includes the MLlib machine learning library. Thanks to its high performance, Spark can do real-time processing as well as batch processing.

Hadoop MapReduce is great for batch processing. If user wants a real-time option you'll need to use another platform like Impala or Apache Storm, and for graph processing you can use Apache Giraph. MapReduce used to have Apache Mahout for machine learning, but it's since been ditched in favor of Spark.

Spark needs a lot of memory. Much like standard databases, Spark loads a process into memory and keeps it there until further notice for the sake of caching. If you run Spark on Hadoop YARN with other resource-demanding services, or if the data is too big to fit entirely into memory, then Spark can have performance degradations. MapReduce, on the other hand, kills its processes as soon as a job is done, so it can easily run alongside other services with minor performance differences.

Spark is a lightning fast cluster computing tool. Apache Spark runs applications up to 100x faster in memory and 10x faster on disk than Hadoop. Because of reducing the number of the read/write cycle to disk and storing intermediate data in-memory spark makes it possible. While MapReduce reads and writes from disk, as a result due to the back and forth, it slows down the processing speed. Now, we will run experiment to prove spark works faster than MapReduce [43].

7.4. Experiment

Now, the listed five queries have run by hive on both the platforms i.e., MapReduce and Spark. And the execution time has noted. The amount of time consumed during input a user query for finding records are listed below:

Table 3: Execution time taken by Hive with Spark

Query	Time in sec
Q1	1.95
Q2	1.65
Q3	1.48
Q4	1.9
Q5	2.00
Q6	4.5
Q7	1.9
Q8	2.6
Q9	3.7
Q10	4.6
Q11	8.7
Q12	6.3
Q13	3.5
Q14	0.95
Q15	2.3

Table 4: Execution time taken by Hive with MapReduce

Query	Time in sec
Q1	9.65
Q2	6.95
Q3	7.58
Q4	13.9
Q5	10.90
Q6	6.7
Q7	3.3
Q8	4.2
Q9	4.7
Q10	7.9
Q11	11.9
Q12	7
Q13	3.00
Q14	1.9
Q15	4.6

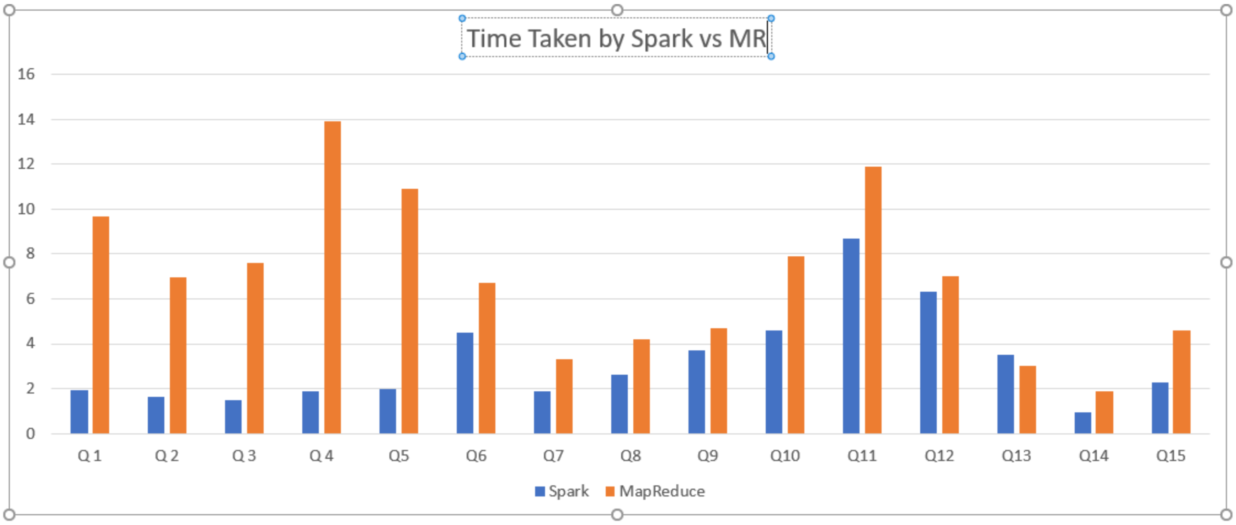


Fig 13: Time in seconds taken by Spark vs MR

7.4.1. Result

It is pretty evident that spark preforms the query execution lot faster than the MapReduce. Since, spark does the in-memory processing whereas MapReduce needs to read and write from the disk. Hence, the speed of processing is different. The graph given below will give the better explanation of the execution speed between spark and MapReduce.

7.5. Hive VS PIG

Hive is an integral part of Hadoop ecosystem which can be used for structured data. Hence, firstly data structure needs to be made before hive table can be injected. As Hive is much familiar to SQL. We can optimize hive query similar to SQL. Moreover, hive contains some additional features including partitioning and bucketing which makes the data analysis easy and quick.

Hive was developed by Facebook and later it became the top apache project which gives the user flexibility by writing less code. It converts the queries into MapReduce execution through developers don't have to worry much about the backend much. Hive uses a query language pretty much similar to SQL known as HQL (Hive query language).

Hive gives option to create UDFs (user-defined function) if something is not available. That will definitely does the work. In short, we can summarize Apache Hive as follows-

- Hive is a data warehouse infrastructure.
- Hive uses a language called HQL, and it is quite similar to SQL.
- It offers various tools for easy extraction, transformation and loading of data.
- Hive allows user to define custom mapper and reducer.
- For data analytics and reporting related work, it is most preferred.

Pig was developed in 2006 by Yahoo to reduce the code complexity with MapReduce. It uses a simple language called Pig Latin as a high-level data flow system specifically used for data manipulation and queries.

Moreover, to store the data schema needs to be created in Pig. It also offers the directly load of the files and start using it.

To be more specific, for Big Data Pig is kind of ETL (extract-transform-load). Also, it is quite useful and can handle large datasets. Moreover, it allows multiple query approach to developers. That reduces the data scan iteration. In addition, we can use multiple nested datatypes. Such as Maps, Tuples, and Bags. Also, we use it for the operations like Filter, Pig Join, and Ordering.

However, for the majority of MapReduce related work, there are many companies who use Pig. In short, we can summarize Apache Pig as follows:

- In other words, Pig is a high-level language called Pig Latin
- Basically, those programmers who are familiar with scripting language prefers pig
- Also, to store the data there is no need to create the schema
- Moreover, Pig's compiler translates Pig Latin into sequences of MapReduce programs

Since Hadoop uses MapReduce to process and analyze big data, processing them takes more time than traditional method. Mapreduce was primarily written in Java and lengthy complex codes were written by programmers to process data. This proved to be disadvantage for users who are non-programmers. To overcome this issue, Hive and Pig were introduced.

Initially, it was hard for Facebook to process and analyze big data since not all the employees were not well equipped with high level programming. Hence, they needed a language similar to SQL which is significantly easier compare to any high-level programming language like Java.

Hence, Hive was developed with a vision to include the concepts of tables, columns just like SQL.

Similarly, Yahoo also found it hard to process and analyze big data using MapReduce as not all the employees were well versed with complex java codes. That's why, there was a necessity to process data using a language which was easier than java. Yahoo researchers developed Pig, which was used to process data quickly and easily.

Hive uses a query language called HiveQL which is similar to SQL used by Hive to process and analyze data. It is a declarative language which is exactly similar to SQL. HiveQL works on structured data [44].

Pig Latin is the procedural data flow language used in pig to analyze data, it is similar to SQL but varies greatly. It is used for structured, semi-structured and unstructured data.

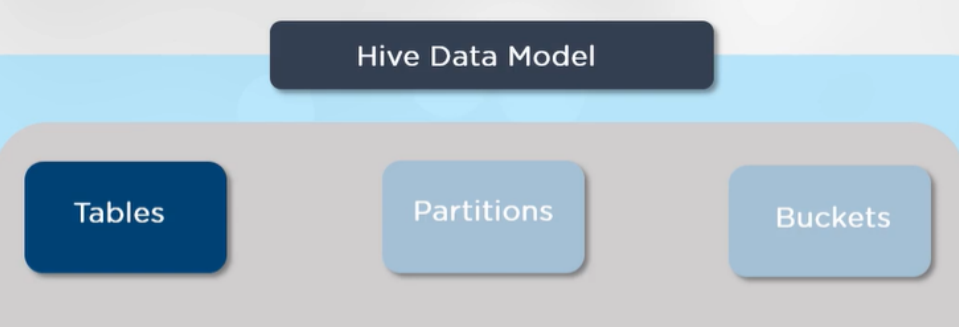


Fig 14: Hive data model

Tables in Hive are similar to those in RDBMS. Tables are grouped into partitions to group the same kind of data based on the partition key. Partitions are further divided into buckets for better query processing.

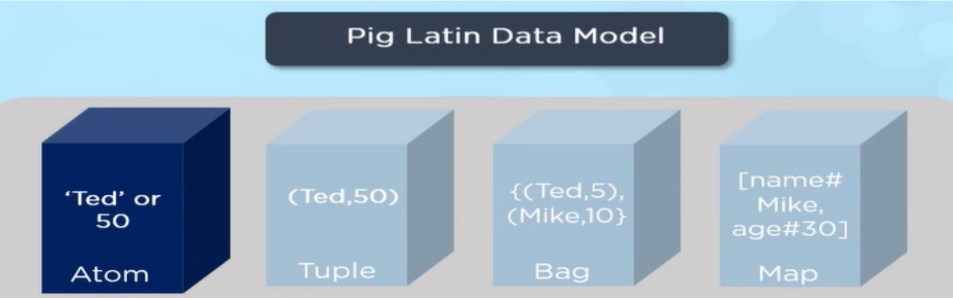


Fig 15: Pig Latin model

- Atom is a single value of primitive data type like int, float, string. It is always stored as string.
- Tuple is a sequence of fields that can be of any data type, it is same as row in RDBMS.
- Bad is a collection of tuples. It is the same as a table in RDBMS. It is represented by '{}'.
- Map is a set of key-value pairs. Keys is of char array type and value can be of any type. It is represented by '[]'

A brief comparison between Hive and Pig is listed below:

Table 5: Comparison between Hive and Pig

Hive	Pig
Used by analysts	Used by programmers and researchers.
HiveQL is the language used.	Pig Latin is the language used.
Only works on structured data.	Structured, semi-structured and untrusted data.
Doesn’t support Avro.	Does supports Avro.
Hive supports partitions.	Pig doesn’t support partitions although there is an option for filtering.
Hive has web interface	Pig doesn’t support web interface.

Avro is a row-oriented remote procedure call and data serialization framework developed within Apache's Hadoop project that uses JSON for defining data types and protocols, and serializes data in a compact binary format. Its primary use is in Apache Hadoop, where it can provide both a serialization and a wire format for communication between Hadoop nodes. These services can be used together or independently. Avro uses a schema to structure the data that is being encoded. It has two different types of schema languages; one for human editing (Avro IDL) and another which is more machine-readable based on JSON.

The amount of time consumed during input a user query for finding records from the hive technique is mentioned as query execution time. In order to measure the query execution time, below listed queries are written on hive interphase and their performance is measured. Below are the list of ten queries that are interpreted both in hive and pigLatin format to measure the execution time [45].

Queries:

1. Select the age of the patients whose cholesterol level is over 200.
2. Select the number of patients who are woman.
3. Select the number of patients who are man.

4. Select the list of banks that filed bankrupt.
5. List of products with price in an ascending order.
6. Select the number of patients who are under 45.
7. List the number of drug victims based on their age.
8. List the gender of the drug victims.
9. List of states with failed bank.
10. List of products with price in a descending order.

Hive SQL:

- Select age from Cleveland where chol>200
- Select count (ID), Sex from Cleveland where Sex='Female'
- Select count (ID), Sex from Cleveland where Sex='Male'
- Select name, ST from banklist where status=' bankrupt'
- Select product_name, price from products order by price ASEC
- Select count (ID) from Patient where age>=45
- Select count (ID), Race from Ace-Drug group by Race.
- Select count (ID), Sex from Ace_Drug Group by Sex.
- Select count (ID), ST from banklist Group by ST
- Select Product_Name, Price From products Order by Price DESC.

Pig Latin:

1. X= FILTER Cleveland by chol >200

   Y= FOREACH age generate X

   DUMP Y;

```
2.  X= FILTER Cleveland by Sex=’Female’

    Y=FOREACH count (ID) generate X

    DUMP Y;

3.  X= FILTER Cleveland by Sex=’Male’

    Y=FOREACH count (ID) generate X

    DUMP Y;

4.  X= FILTER banklist by Status=’ bankrupt’’

    Y=FOREACH name, ST generate X

    DUMP Y;


5.  X= FILTER Products by ProductName, Price

    Y= ORDER X by Price

    Z= ORDER Y by $1 ASEC

    DUMP Z;

6.  X= FILTER Patient by Age >= 45

   Y= FOREACH COUNT (ID) generate X

   DUMP Y;

7.  X= FILTER Ace_Drug by ID, Race

    Y= GROUP X by Race

    OUTPUT= FOREACH Y GENERATE Race, COUNT (ID)

    DUMP OUTPUT;

8.  X= FILTER Ace_Drug by ID, Sex
    Y= GROUP X by Sex
    OUTPUT= FOREACH Y GENERATE Sex, COUNT (ID)
    DUMP OUTPUT;
9.  X= FILTER banklist by ID, ST
    Y= GROUP X by ST
    OUTPUT= FOREACH Y GENERATE ST, COUNT (ID)
    DUMP OUTPUT;
10. X= FILTER Products by ProductName, Price
    Y= ORDER X by Price
    Z= ORDER Y by $1 DESC
    DUMP Z;
```

Table 6: Execution time taken by Hive vs Pig

Query	Hive	Pig
Q1	8.3	13.49
Q2	3.47	5.33
Q3	6.49	9.47
Q4	5.44	8.97
Q5	4.30	7.86
Q6	3.6	6.5
Q7	4.8	5.6
Q8	8.7	13.5
Q9	9.6	14.22
Q10	5.6	8.3

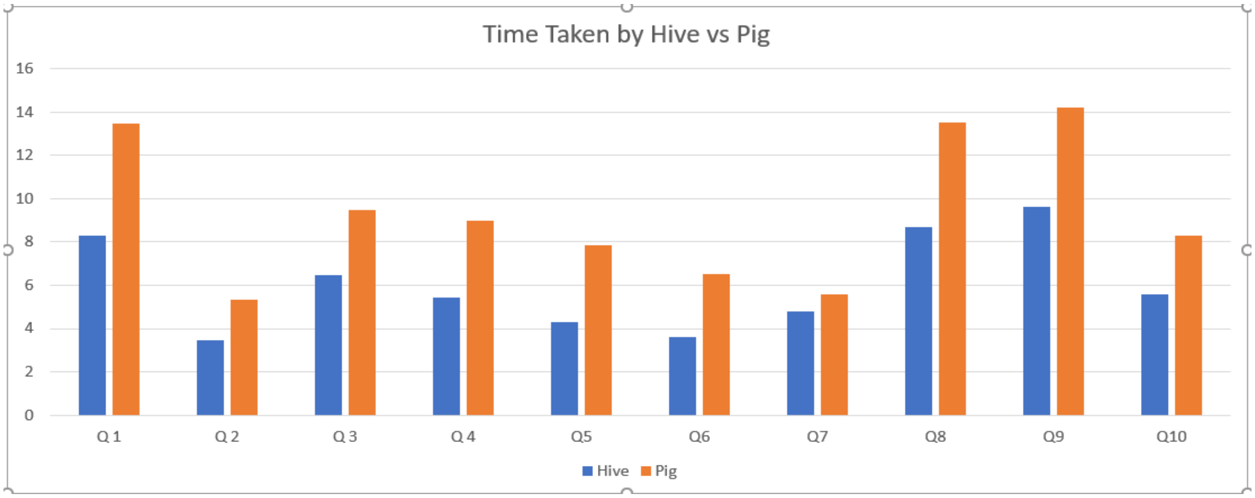


Fig 16: Time taken by Hive vs Pig

8. CONCLUSION

In this paper, I explained many possible details of big data including the types, application, uses, importance and all available tools in the big data technologies. Since, this technology deals with all possible sizes of data especially huge sized data, so I used some big datasets to run the experiments. I made the comparison among tools for example spark vs mapReduce, Sqoop vs Flume, Tez vs mapReduce, Pig vs Hive etc. Not only that, this paper also answers the critical questions like why each of these tools are faster than their contemporary one and to answer these questions, I had to run different experiments using different kinds of datasets. And, the comparative study was interpreted in graphical manner for the better understanding of the differences. Lastly in future, I want to work on big data to bring answer with more precision.

REFERENCES

1. S. K. Sahu, M. M. Jacintha and A. P. Singh, "Comparative study of tools for big data analytics: An analytical study," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 37-41, doi: 10.1109/CCAA.2017.8229827.

2. Ikhlaq, S. & Ikhlaq, Sheikh. (2017). A comparative study of big data computational approaches. International Journal of Applied Engineering Research. 12. 8131-8136.

3. Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. Health information science and systems, 2, 3. https://doi.org/10.1186/2047-2501-2-3

4. Munné R. (2016) Big Data in the Public Sector. In: Cavanillas J., Curry E., Wahlster W. (eds) New Horizons for a Data-Driven Economy. Springer, Cham. https://doi.org/10.1007/978-3-319-21569-3_11

5. Silva, E., Hassani, H., & Madsen, D. (2019). Big Data in fashion: transforming the retail sectorJournal of Business Strategy, ahead-of-print.

6. O'Donovan, P., Leahy, K., Bruton, K. et al. Big data in manufacturing: a systematic mapping study. Journal of Big Data 2, 20 (2015). https://doi.org/10.1186/s40537-015-0028-x

7. Lee, J.G., & Minseo, K. (2015). Geospatial Big Data: Challenges and OpportunitiesBig Data Research, 2.

8. Diego Matricano, Framing the entrepreneurship phenomenon, Entrepreneurship Trajectories, 10.1016/B978-0-12-818650-3.00001-5, (1-31), (2020).

9. Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challengesBusiness Horizons, 60.

10. Balachandran, B., & Prasad, S. (2017). Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business IntelligenceProcedia Computer Science, 112, 1112-1122.

11. Big data defined. (n.d.). Www.Oracle.Com/. Retrieved November 4, 2020, from https://www.oracle.com/big-data/what-is-big-data.html

12. Balachandran, Bala M., and Shivika Prasad. "Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business Intelligence." Procedia Computer Science, vol. 112, Jan. 2017, pp. 1112–22. ScienceDirect, doi:10.1016/j.procs.2017.08.138.

13. Savaram, R. (n.d.). Apache Sqoop vs Apache Flume. Https://Mindmajix.Com/. Retrieved November 4, 2020, from https://mindmajix.com/apache-sqoop-vs-apache-flume.

14. Salloum, S., Dautov, R., Chen, X., Peng, P. X., & Huang, J. Z. (2016a). Big data analytics on Apache Spark. International Journal of Data Science and Analytics, 1(3–4), 145–164. https://doi.org/10.1007/s41060-016-0027-9

15. Farhan, Md. Nowraj & Habib, Md. Ahsan & Ali, Arshad. (2018). A study and Performance Comparison of MapReduce and Apache Spark on Twitter Data on Hadoop Cluster. International Journal of Information Technology and Computer Science. 10. 61-70. 10.5815/ijitcs.2018.07.07.

16. Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Research in Computer Science, 1–10. https://doi.org/10.9734/ajrcos/2019/v4i230108

17. Vyawahare, H. R., Karde, D. P. P., & Thakare, D. V. M. (2017). Brief Review on SQL and NoSQL. International Journal of Trend in Scientific Research and Development, Volume-2(Issue-1), 968–971. https://doi.org/10.31142/ijtsrd7105

18. Ali, W., Majeed, M., Raza, A., & Shafique, M. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data AnalyticsAsian Journal of Computer Science and Information Technology, 4, 1-10.

19. Moniruzzaman, A., & Hossain, S. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. ArXiv, abs/1307.0191.

20. Băzăr, C., & Iosif, C.S. (2014). The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase. Database Systems Journal, 5, 49-59.

21. Thomas Mason, R. (2015).NoSQL Databases and Data Modeling for a Document-oriented NoSQL Database. Proceedings of the 2015 InSITE Conference, 260–267. https://doi.org/10.28945/2245.

22. Pothuganti, A. (2015). Big Data Analytics : Hadoop-Map Reduce & NoSQL Databases.

23. Marr, B. (2016). Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results (1st ed.). Wiley.

24. Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Research in Computer Science, 1–10. https://doi.org/10.9734/ajrcos/2019/v4i230108

25. Nayak, A. (2013). Type of NOSQL Databases and its Comparison with Relational Databases.

26. A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, & U. Saxena (2017). NoSQL databases: Critical analysis and comparison. In 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN) (pp. 293-299).

27. Tudorică, B., & Bucur, C. (2011). A comparison between several NoSQL databases with comments and notes. 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research, 1-5.

28. Cherifi, D., Radji, N., & Nait-Ali, A. (2011). "Effect of Noise Blur and Motion on Global Appearance Face Recognition based Methods Performance." International Journal of Computer Applications, 16(6), 4–13. https://doi.org/10.5120/2019-2723.

29. Horne, J. (2018). Visualizing Big Data From a Philosophical Perspective. Advances in Data Mining and Database Management, 809–852. https://doi.org/10.4018/978-1-5225-3142-5.ch028.

30. Jha, Meena & Jha, Sanjay & O'Brien, Liam. (2016). Combining big data analytics with business process using reengineering. 1-6. 10.1109/RCIS.2016.7549307.

31. Akter, S., & Wamba, S. F. (2016). Big data analytics in E-commerce: a systematic review and agenda for future research. Electronic Markets, 26(2), 173–194. https://doi.org/10.1007/s12525-016-0219-0

32. Aswani, R., Kar, A. K., Ilavarasan, P. V., & Dwivedi, Y. K. (2018). Search engine marketing is not all gold: Insights from Twitter and SEOClerks. International Journal of Information Management, 38(1), 107–116. https://doi.org/10.1016/j.ijinfomgt.2017.07.005.

33. Wang, Y., & Hajli, N. (2017). Exploring the path to big data analytics success in healthcare. Journal of Business Research, 70, 287–299. https://doi.org/10.1016/j.jbusres.2016.08.002

34. Karaboga, T. (2019). Big Data Analytics And Firm Innovativeness: The Moderating Effect Of Data-Driven Culture. Big Data Analytics And Firm Innovativeness: The Moderating Effect Of Data-Driven Culture, 526–535. https://doi.org/10.15405/epsbs.2019.01.02.44

35. Lopez-Nicolas, C., & Soto-Acosta, P. (2010). Analyzing ICT adoption and use effects on knowledge creation: An empirical investigation in SMEs. International Journal of Information Management, 30(6), 521–528. https://doi.org/10.1016/j.ijinfomgt.2010.03.004

36. Rehman, M. H., Chang, V., Batool, A., & Wah, T. Y. (2016). Big data reduction framework for value creation in sustainable enterprises. International Journal of Information Management, 36(6), 917–928. https://doi.org/10.1016/j.ijinfomgt.2016.05.013

37. Grant, R. M. (2013). Nonaka's 'Dynamic Theory of Knowledge Creation' (1994): Reflections and an Exploration of the 'Ontological Dimension.' Towards Organizational Knowledge, 77–95. https://doi.org/10.1057/9781137024961_5

38. Ruths, D., & Pfeffer, J. (2014). Social media for large studies of behavior. Science, 346(6213), 1063–1064. https://doi.org/10.1126/science.346.6213.1063

39. Fan, W., & Gordon, M. D. (2014). The power of social media analytics. Communications of the ACM, 57(6), 74–81. https://doi.org/10.1145/2602574

40. Fischer, M., & Himme, A. (2017). The financial brand value chain: How brand investments contribute to the financial health of firms. International Journal of Research in Marketing, 34(1), 137–153. https://doi.org/10.1016/j.ijresmar.2016.05.004

41. Lawrence, J. M., Crecelius, A. T., Scheer, L. K., & Patil, A. (2019). Multichannel Strategies for Managing the Profitability of Business-to-Business Customers. Journal of Marketing Research, 56(3), 479–497. https://doi.org/10.1177/0022243718816952