Big Data Engineering Technologies

Lwando Maciti
Computer Science Department
University of the Western Cape
Bellville, Cape Town, South Africa
3756147@myuwc.ac.za

***Abstract*- Access to the internet using technological devices has become easier than it was before. As a result, large amounts of data are being produced daily from different technological platforms and devices such as smartphones, laptops, smartwatches, and other IoT devices. These large amounts of data are often unstructured and meaningless. However, it becomes of high value once it has been processed and analyzed. The relevant organizations may find value from the trends and patterns identified from the data – what they do with theses analysis is what gives the data real meaning and hence the term "Big Data". This paper aims to discuss the different ways of producing large amounts of data and its challenges and how big data technologies can be used to process and analyze large amounts of data to obtain value from it.**

Keywords—Big Data, Big Data Analytics, Data Engineering

I. INTRODUCTION

Everyone owning or using a technological device produces data even without realizing it. For example, we produce data from social media posts, electronic transactions, sensor devices such as temperature scanners, and even from the pictures we take on our phones. This data can either be very useful or of no use at all, depending on what problem the data is going to be used to solve.

There are various fields where data can be used, such as Machine Learning, where data is used to develop classification or predictive models and in Business Intelligence which involves the descriptive analysis on how the business is performing and how it has previously performed [1]. Big data is not just big data because it consists of large amounts of data, which gives it its essence is the value it provides from the analysis made on it. Data can be a very important asset in an organization as it can help them identify areas of improvement which consequently improves business productivity. Big data analysis generally improves decision making within organizations or an individual. Data seems to be becoming an asset in businesses, so much that they are investing in improving their data infrastructures [2].

Big data engineering can be formally described as the systematic process of ensuring that data is made available to its stakeholders when required. It is a newly formed field with its focus being "the design, implementation, and maintenance of distributed information systems" [2]. Big data is often produced in such large amounts of quantities that it cannot be captured, stored, and analyzed using normal databases [3]. Hence, advanced systems need to be engineered to facilitate the capturing, storage, and analysis of this data. To do so, the nature of the data needs to be understood, as well as the purpose it will be serving should be clearly defined. As a result, technologies such as Hadoop and many others have been developed to facilitate the storage and processing of big data.

II. BACKGROUND

The main objective of engineering big data is to ensure that all data needs required by the information systems have been addressed. Big data engineering is a guided process consisting of the following important components, data handling a representation, data architecture (expert systems), data construction, handling issues concerned with application and management, tools for specifying and developing data and implementation, and design [2].

Data analysis is regarded as an important tool in data engineering, it makes use of statistical models and techniques to evaluate data [2]. Data analysis also helps to establish important relationships found in the data, it allows for comparisons between data variables to be made and allows us to predict possible future events or forecasting [4]. Figure 1 below outlines the steps to follow when performing data analysis.

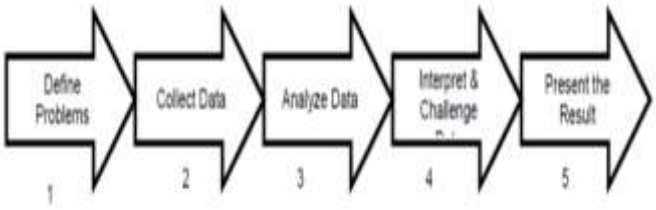

Figure 1 Data analysis procedure [2].

A. *Define the problem*

This is where you clearly define the nature and scope of the problem you aim to solve using big data engineering, this will also help in identifying ways of collecting the required data.

B. *Collect data*

Identify the data collection source suitable for the problem you want to solve, for example, social media posts, temperature sensors et cetera and start the data collection process.

C. *Analyze the data*

Perform data cleansing and processing and develop statistical models to identify any trends and important patterns in the data.

D. *Interpret the information*

Add meaning to the data by explaining and drawing conclusions about the identified trends and patterns mean.

E. *Present the results*

This consists of the written reports containing the visualization of the results, also making note of all the

limitations of the experiment and the conclusions you can and cannot make [2][4].

III. THREE FORMS OF BIG DATA

A. *Structured Data*

The structure of this type of data is well defined. The data types and structure are predefined in a schema, and all the records are split into corresponding rows and columns. This type of data is easy to read and manage – it is normally presented in a relational database. The main sources of this data are usually humans and machines, data can be manually added into the database system by its users. However, this type of data "accounts for only 20% of the total available data" [5][6]. Figure 2 below is a typical example of a database schema and a table in a database system.

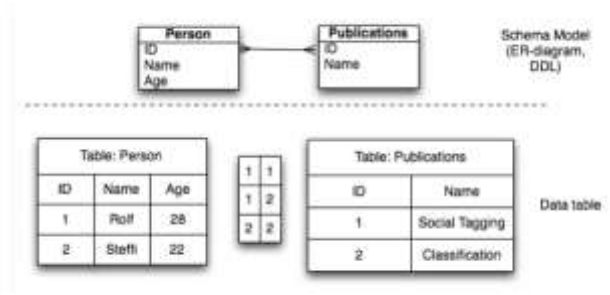

Figure 2 Database table and schema example [6].

B. *Semi-structured Data*

This type of data is not predefined in a schema; however, it is possible to define it in the form of a schema if required. It is not so different from structured data; the main distinguishing factor is that semi-structured data cannot be stored in a relational database. This type of data also allows for variations in its structure, for example, it can contain duplicate and there can also exist minor changes to the structure. CSV files and no SQL are popular examples of semi-structured data [5][6]. Figure 3 below shows a typical example of an RDF graph which represents unstructured data.

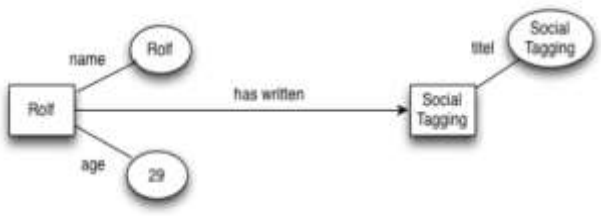

Figure 3 RDF Graph example [6].

C. *Unstructured Data*

Unstructured data is data that has no defined structured or format of storage. A traditional database system can not be used as a means of storing this kind of data. Any kind of data can be stored in unstructured data since it has no restrictions on the format or structure of the data. However, this limits the constraints that can be applied to the data, this also adds complexity to the storage and management of the data [5][6]. A typical example of this type of data could be data retrieved from twitter posts. Different forms of data can be retrieved from a twitter post, for example, images, text, video, and the number of likes and retweets. Table 1 below represents the growth rate of unstructured data produced by different companies according to [7].

TABLE 1 Growth rate of unstructured data:

Company	Generated Data
Digital Study	1,227 Exabyte in 2010
	Predicted 1.8 Zettabyte data creation annually in 2011
	2.7 Zettabytes in 2013
YouTube	48 hours of new video uploaded every minute
Facebook	34,722 Likes received every minute
	100 terabytes uploaded daily
	30+ Petabytes (stores, accesses, and analyzes) 30 Billion Pieces of content shared monthly
Domain Name	571 new websites are created every minute
Web store	More than 2.5 petabytes hourly
Twitter	Roughly 175 million tweets every day
	More than 465 million accounts
Boeing 787	40 terabytes (TB) per hour of flight
Oil drilling	Up to 2.4 TB per minute
Automated manufacturing facility	Approximately 1 TB per hour
Large retail store	Approximately 10 gigabytes (GB) per hour
Global data center IP traffic	8.6 Zettabytes annually
Universe Data generated by IoE	400 ZB
The world	Creates 2.5 quintillion bytes of data per day

Figure 4 Growth of unstructured data [7].

IV. MANAGING BIG DATA USING BIG DATA TOOLS

As we can see from table 1 that data is rapidly growing from the extensive production of data from the different platforms. This data is too large to store and manage using traditional database systems [3]. This may be because the current computer hardware may not be able to handle a large amount of data processing and manipulation. This leads to the need of developing advanced systems that can handle computations of large amounts of data. The Apache Hadoop software was developed as a possible solution to this problem.

A. *Hadoop*

Hadoop is an open-source software developed by Apache to solve the problem of processing large amounts of data by providing a "reliable shared storage and analysis system". [8]. Hadoop consists of two main components, Hadoop File Systems (HDFS) which handles data storage and MapReduce which provides the analysis of the data [8][9].

The downside about Hadoop is that it is more suitable for batch jobs, which is the processing of data that has already been captured, therefore, it is not suitable for live data streaming or interactive data analysis [10].

B. *HDFS*

Hadoop File Systems is a distributed file system used by Hadoop for reliable storage and access to large volume data. "It consists of blocks, name node (master) and data node (slave)" [11]. These blocks indicate the minimum amount of data that it can perform a read or write operation [11].

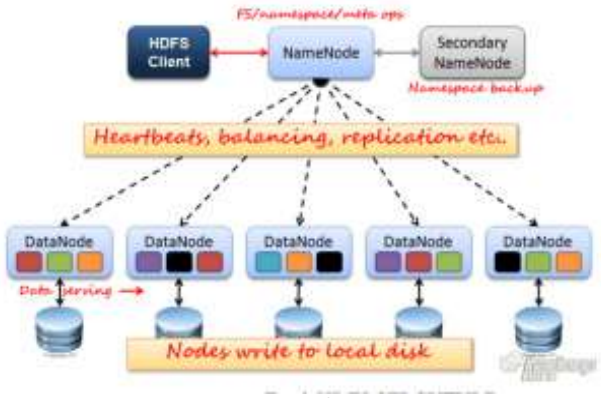

Figure 5 Structure of an HDFS [11].

Some of the pros of HDFS are that it can contain high bandwidth to support map-reduce operations. It is inexpensive and data can be written once and read several times. The cons of HDFS are that it is quite complex to manage the different clusters and joining multiple databases takes a lot of computational time [11].

C. *MapReduce*

Map-reduce is a tool used by Hadoop for the parallel processing of large amounts of data in many clusters. It consists of two nodes, the Job tracker node, and the Task tracker node. The job tracker node allocates tasks and resources to the task node, the task node then executes the allocated tasks. "It contains two important functions, the MAP() and the REDUCE() function" [11].

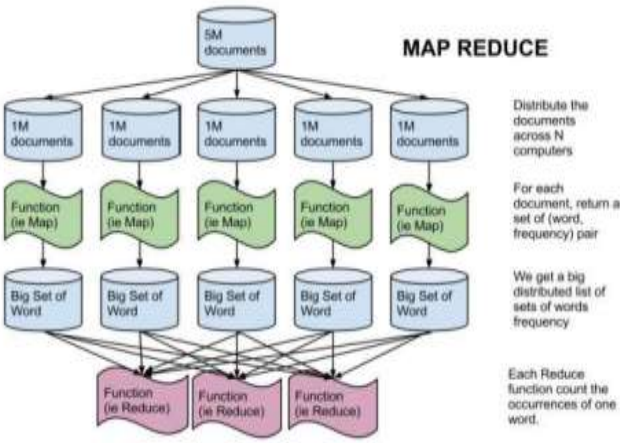

Figure 6 Map Reduce structure [11].

The following are the two map-reduce functions:

- *MAPPER ()*
  This function takes grouped data and divides it into multiple splits and individual elements are broken into key/value tuples [11].
- *REDUCER ()*
  It retrieves the input from the map () function and joins it the data tuples into smaller tuples. This function initiates after the map () function to produce the final output [11].

Map-reduce supports various programming languages including java. It cannot be used for intensive interactive data analysis and it only works for batch processing [11].

D. *SQOOP*

The main function of the Sqoop framework is to coordinate the transfer of data between Hadoop and a database system in a parallel manner. It "refers to SQL to Hadoop" and vice versa. The ETL processes, "Extraction, Transformation and Loading can be performed using Sqoop". Sqoop provides the movement of a mixture of data with easy integration. It can import a single table in a database up to an entire database into HDFS [11].

V. REAL-TIME DATA PROCESSING (STREAM-PROCESSING)

The Hadoop framework is more suitable for batch jobs or batch processing since its map-reduce function does not support stream processing, this means that it cannot provide real-time data analytics. This indicates a need for real-time (non-batch processing) data processing techniques [10]. Therefore, we will be looking at stream processing as a means of obtaining real-time data processing or interactive data analysis.

Data streams refer to data that is continuously generated at any rate. It is characterized by the "continuous arrival of data objects, unordered arrival of data objects, and an unbounded size of a stream [12]. Typical examples could be twitter likes, clickstreams, and message streams [10].

Map-reduce does not support streaming, however, by treating the batches of data as small chunks allow for map-reduce to handle streaming through a process called micro-batching. Map-reduce implementations such as Spark support this method of streaming through a process "called discretized stream or DStream" [10]. Data streams are applicable in several scenarios for example, in continuous queries, where queries are made on data streams to monitor changes in data for the continuous response to queries. Datastream mining can also obtain valuable information from stream data [12].

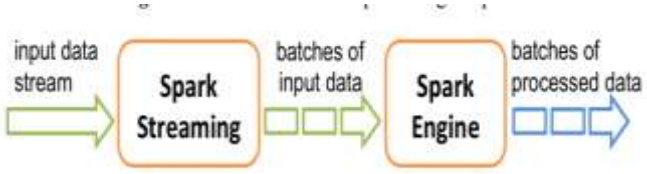

Figure 7 Stream processing schema with Spark [10].

VI. CHALLENGES WITH BIG DATA

Datastores are becoming large and data engineering is facing the difficulty of simplifying the access and management of the data [2]. The extensive production of data gives rise to the high variability in the data. Data is now produced in many different types and formats as a result of the increase in usage of smart devices and the Internet of Things (IoT). The high speed of data production exceeds the capacity of current software and, consequently, these software systems are not able to handle the processing of these large amounts of data. Since the data is generated from various sources, there could be issues of misspelling or misclassification which could make the data inaccurate and consequently lowers the overall quality of the data [13]. Other issues to be concerned with are, privacy issues where the use of personal data is applicable. Data security is another challenge; data needs to be protected and secured to prevent the damage that may be caused when it lands on the wrong hands.

VII. CONCLUSION

The different forms of big data have been outlined and the examples thereof. From that we can deduct that, big data is mostly generated from unstructured data from different smart or IoT devices. We also revealed that stream processing is a more useful way of processing data as it provides the ability to interactively analyze and query data. Big data can be applied in many fields, including machine learning, Business Intelligence, and data mining. In as much as the idea of big data engineering is exciting, the extensive production of data leads to several challenges such as data quality issues, data processing issues, and data volume challenges. Lastly, an important thing to note about big data is that big data is not just big data on its own, but what makes it big data is the processing and analysis you do with it.

REFERENCES

[1] M. Niño and A. Illarramendi, “Entendiendo El Big Data: Antecedentes, Origen Y Desarrollo Posterior,” *Dyna New Technol.*, vol. 2, no. 3, p. [8 p.]-[8 p.], 2015, doi: 10.6036/nt7835.

[2] K. Eze, “The Essence of Data Engineering,” no. November, pp. 2–4, 2018.

[3] L. Wang and C. A. Alexander, “Big data in design and manufacturing engineering,” *Am. J. Eng. Appl. Sci.*, vol. 8, no. 2, pp. 223–232, 2015, DOI: 10.3844/ajeassp.2015.223.232.

[4] “Data analysis, interpretation, and presentation,” *Use 137Cs Soil Eros. Assess.*, pp. 34–39, 2019, DOI: 10.18356/a2ceb52f-en.

[5] P. Tiwari, “Comparative Analysis of Big Data,” *Int. J. Comput. Appl.*, vol. 140, no. 7, pp. 24–29, 2016, DOI: 10.5120/ijca2016909400.

[6] R. Sint, S. Schaffert, S. Stroka, and R. Ferstl, “Combining unstructured, fully structured and semi-structured information in semantic wikis,” *CEUR Workshop Proc.*, vol. 464, no. May, pp. 73–87, 2009.

[7] A. C. Eberendu, “Unstructured Data: an overview of the data of Big Data,” *Int. J. Comput. Trends Technol.*, vol. 38, no. 1, pp. 46–50, 2016, DOI: 10.14445/22312803/ijctt-v38p109.

[8] S. Blazhievsky, “Introduction to Hadoop, MapReduce, and HDFS for Big Data Applications,” p. 67, 2013.

[9] P. J. Charles, S. T. Bharathi, and V. Susmitha, “Big Data – Concepts, Analytics, Architectures – Overview,” *Int. Res. J. Eng. Technol.*, vol. 5, no. 2, pp. 125–129, 2018.

[10] S. Shahrivari, “Beyond batch processing: Towards real-time and streaming big data,” *Computers*, vol. 3, no. 4, pp. 117–129, 2014, DOI: 10.3390/computers3040117.

[11] S. Suguna, “Improvement of HADOOP Ecosystem and Their Pros and Cons in Big Data,” *Int. J. Eng. Comput. Sci.*, no. May 2016, 2016, DOI: 10.18535/ijecs/v5i5.57.

[12] D. Namiot, “On Big Data Stream Processing,” *Int. J. Open Inf. Technol.*, vol. 3, no. 8, pp. 48–51, 2015, [Online]. Available: http://injoit.org/index.php/j1/article/view/225.

[13] Q. Maqbool and A. Habib, “5Big Data challenges,” *Control Eng.*, vol. 66, no. 3, p. 33, 2019, DOI: 10.4172/2324-9307.1000133.