# Challenges and Research Disputes and Tools in Big Data Analytics

**K. Venkatesh, M. Jafar Sathick Ali, N.Nithiyanandam, M. Rajesh**

***Abstract - Big Data is the era of data processing. Big Data is the Collate's observer data sets that are complicated that traditional data-processing abilities.There are the various challenges include data analysis, capture the data, curation, search, sharing, stowage, transmission, visualization, and privacy violations. A large collections of petabytes of data is engendered day by day from the up-to-date information systems and digital era such as Internet of Things and cloud computing. Big data environs is used to attain, organize and analyse the numerous types of data. A large scale distributed file system which should be a fault tolerant, flexible and scalable. The term big data comes with the new challenges to input, process and output the data The technologies used by big data application to handle the massive data are Hadoop, Map Reduce, Pig, Apache Hive, No SQL and Spark. Initially, we extant the definition of big data and discuss big data challenges. Succeeding, The Propionate Paramour of Big Data Systems Models in the Into Prolonging Seam, Namely data Generation, data Assange, data Storage, and data Analytics. These four modules form a big data value chain. In accumulation, we present the prevalent Hadoop framework for addressing big data.***

***Keywords— Big Data, Hadoop, Map Reduce, Apache Hive, No SQL and Spark.***

## I. INTRODUCTION

At the end of the everyday, it is about size. In this world where size matters, big data became really a big and valuable term. Data's are generated from various sources and the fast transition from digital technologies has led to growth of big data. [4, 6] This data could be structured, unstructured or Semi-structured. There is a need to convert Big Data into Business Intelligence that enterprises can readily deploy. Better data leads to better decision making and an improved way to strategize for organizations regardless of their size, geography, market share, customer satisfaction, customer segmentation and such other categorizations. There are the various tools are available to analytics the data. Hadoop is one of the tool for working with extremely large volumes of data. [1, 3] The V's of Big Data (volume, velocity, variety, veracity, valence, and value) and why each impacts data collection, monitoring, storage, analysisand reporting.

**Revised Manuscript Received on 14 August, 2019.**

**K.Venkatesh**,Assistant Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamilnadu, India.

**M. Jafar SathickAli**,Assistant Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamilnadu, India.

**N.Nithiyanandam**,Assistant Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamilnadu, India.

**M.Rajesh,** Dept of Computer Science and Engineering, KRS College of Engineering, India., RaGa Academic Solutions, Chennai, India

**Volume:**

It refers to the large amount of data involved with big data. The scale of datasets keeps accumulative from MB→EB

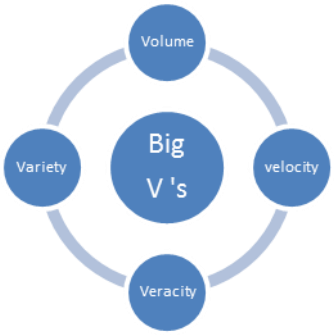


**Fig. 1.1: Parameters of Big Data**

The **Variety:**

The Variety which refers the various types of data's like pdf, Doc, Mp3. It based the structured, semi-structured and unstructured data [13].

**Veracity:**

The increase in the range of values typical of a large data set. When we dealing with high volume, velocity and variety of data it mainly indicates the trustiness of the data.

**Velocity:**

The data comes at high speed. It terms with the time management. The data's are used in the mainframe systems, client-server model, Internet, Mobile & Cloud infrastructure.

## II. BIGDATA - COLLECTIONS

Big data involves the data produced by different devices and applications [13]. These are some of the fields which can generate big data.

- **Social Sites Data:** Social media such as Facebook, Twitter, etc. [7, 5]. Carry information, suggestions, invitations, etc. posted by several people across the world. The responses for their campaigns, advertising mediums, etc are also known.
- **Search Engine Data:** Search engines retrieve lots of data from different databases.
- **Medical History Data:** Hospitals can generate medical history of patients for various health issues [9, 10].
- **Online Shopping Data:** Shopping of various products online can help to know thepreferences and product perception of the customers on different products at different intervals.

- **Stock Exchange Data:** The stock exchange data holds information about the shares of various companies. These data given an insight on the decisions taken by shareholders forthe trading activities.
- **Vehicle Booking Data:** Booking of vehicles like train, bus, flight, cab, etc. can generate the data of booking a vehicle based on model, size, distance and availability of a vehicle.
- **Aviation Data:** Audio and Video data recordings, the performance information of the aircraft, etc.

Big Data refers to datasets whose size is beyond the capability of typical database software tools to capture, store, manage, and analyze.

**McKinsey.**

Big Data usually includes datasets with sizes beyond the capability of commonly used software tools to capture, curate, manage, and process the data within a tolerable elapsed time.

**Wikipedia.**

Big Data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery, and process optimization.

## III. HOW IS BIG DATA ANALYSED?

One of the best-known methods for turning raw data into useful information is what is known as MapReduce. MapReduce is a method for taking a large data set and performing computations on it across multiple computers, in parallel. It serves as a model for how to program and is often used to refer to the actual implementation of this model. It is difficult to do the process due to proper application of data in map reduction technique [2]. In essence, MapReduce consists of two parts. The Map function does sorting and filtering, taking data and placing it inside of categories so that it can be analyzed [8]. The Reduce function provides a summary of this data by combining it all together. While largely credited to research that took place at Google, MapReduce is now a generic term and refers to a general model used by many technologies.

## IV. COMPUTING AND ANALYZING DATA

The system can begin processing the data to surface actual information. The computation layer is perhaps the most diverse part of the system as the requirements and best approach can vary significantly depending on what type of insights desired. Data is often processed repeatedly, either iteratively by a single tool or by using a number of tools to surface different types of insights.

Batch processing is one method of computing over a large dataset. The process involves breaking work up into smaller pieces, scheduling each piece on an individual machine, reshuffling the data based on the intermediate results, and then calculating and assembling the final result. These steps are often referred to individually as splitting, mapping, shuffling, reducing, and assembling, or collectively as a distributed map reduce algorithm. This is the strategy used by Apache Hadoop's MapReduce. Batch processing is most useful when dealing with very large datasets that require quite a bit of computation.

While batch processing is a good fit for certain types of data and computation, other workloads require more real-time processing. Real-time processing demands that information be processed and made ready immediately and requires the system to react as new information becomes available. One way of achieving this is stream processing, which operates on a continuous stream of data composed of individual items. Another common characteristic of real-time processors is in-memory computing, which works with representations of the data in the cluster's memory to avoid having to write back to disk [11].

Apache Storm, Apache Flink, and Apache Spark provide different ways of achieving real-time or near real-time processing. There are trade-offs with each of these technologies, which can affect which approach is best for any individual problem. In general, real-time processing is best suited for analyzing smaller chunks of data that are changing or being added to the system rapidly.

The above examples represent computational frameworks. However, there are many other ways of computing over or analyzing data within a big data system. These tools frequently plug into the above frameworks and provide additional interfaces for interacting with the underlying layers. For instance, Apache Hive provides a data warehouse interface for Hadoop, Apache Pig provides a high level querying interface, while SQL-like interactions with data can be achieved with projects like Apache Drill, Apache Impala, Apache Spark SQL, and Presto. For machine learning, projects like Apache SystemML, Apache Mahout, and Apache Spark's MLlib can be useful. For straight analytics programming that has wide support in the big data ecosystem, both R and Python are popular choices

## V. TOOLS USED TO ANALYZE BIG DATA

Perhaps the most influential and established tool for analyzing big data is known as Apache Hadoop. Apache Hadoop is a framework for storing and processing data at a large scale, and it is completely open source. Hadoop can run on commodity hardware, making it easy to use with an existing data centre, or even to conduct analysis in the cloud. Hadoop is broken into four main parts:

- The Hadoop Distributed File System (HDFS), which is a distributed file system designed for very high aggregate bandwidth;
- YARN, a platform for managing Hadoop's resources and scheduling programs that will run on the Hadoop infrastructure;
- MapReduce, as described above, a model for doing big data processing;
- And a common set of libraries for other modules to use.

**Other Big Data Tools**

There are countless open source solutions for working with big data, many of them specialized for providing optimal features and performance for a specific niche or for specific hardware configurations.

The Apache Software Foundation (ASF) supports many of these big data

projects. Here are some that you may find useful [19, 21].

- Apache Beam is "a unified model for defining both batch and streaming data-parallel processing pipelines." It allows developers to write code that works across multiple processing engines.
- Apache Hive is a data warehouse built on Hadoop. A top-level Apache project, it "facilitates reading, writing, and managing large datasets … using SQL."
- Apache Impala is an SQL query engine that runs on Hadoop. It's incubating within Apache and is touted for improving SQL query performance while offering a familiar interface.
- Apache Kafka allows users to publish and subscribe to real-time data feeds. It aims to bring the reliability of other messaging systems to streaming data.
- Apache Lucene is a full-text indexing and search software library that can be used for recommendation engines. It's also the basis for many other search projects, including Solr and Elasticsearch.
- Apache Pig is a platform for analyzing large datasets that runs on Hadoop. Yahoo, which developed it to do MapReduce jobs on large datasets, contributed it to the ASF in 2007.
- Apache Solr is an enterprise search platform built upon Lucene.
- Apache Giraph is an iterative graph processing system.
- Algorithms.io provides predictive analytics for streaming machine-generated data
- Apache Zeppelin is an incubating project that enables interactive data analytics with SQL and other programming languages.

Other open source big data tools you may want to investigate include:

- Elasticsearch is another enterprise search engine based on Lucene. It's part of the Elastic stack (formerly known as the ELK stack for its components: Elasticsearch, Kibana, and Logstash) that generates insights from structured and unstructured data.
- Cruise Control was developed by LinkedIn to run Apache Kafka clusters at large scale.
- TensorFlow is a software library for machine learning that has grown rapidly since Google open sourced it in late 2015. It's been praised for "democratizing" machine learning because of its ease-of-use.

As big data continues to grow in size and importance, the list of open source tools for working with it will certainly continue to grow as well

## VI. VISUALIZING THE RESULTS

Due to the type of information being processed in big data systems, recognizing trends or changes in data over time is often more important than the values themselves. Visualizing data is one of the most useful ways to spot trends and make sense of a large number of data points.

Real-time processing is frequently used to visualize application and server metrics. The data changes frequently and large deltas in the metrics typically indicate significant impacts on the health of the systems or organization. In these cases, projects like Prometheus can be useful for processing the data streams as a time-series database and visualizing that information.

One popular way of visualizing data is with the Elastic Stack, formerly known as the ELK stack. Composed of Logstash for data collection, Elasticsearch for indexing data, and Kibana for visualization, the Elastic stack can be used with big data systems to visually interface with the results of calculations or raw metrics. A similar stack can be achieved using Apache Solr for indexing and a Kibana fork called Banana for visualization. The stack created by these is called Silk.

Another visualization technology typically used for interactive data science work is a data "notebook". These projects allow for interactive exploration and visualization of the data in a format conducive to sharing, presenting, or collaborating. Popular examples of this type of visualization interface are Jupyter NotebookandApache Zeppelin**.**

### Big Data Glossary

While we've attempted to define concepts as we've used them throughout the guide, sometimes it's helpful to have specialized terminology available in a single place [14, 15, 17]:

- **Big data**: Big data is an umbrella term for datasets that cannot reasonably be handled by traditional computers or tools due to their volume, velocity, and variety. This term is also typically applied to technologies and strategies to work with this type of data.
- **Batch processing**: Batch processing is a computing strategy that involves processing data in large sets. This is typically ideal for non-time sensitive work that operates on very large sets of data. The process is started and at a later time, the results are returned by the system.
- **Cluster computing**: Clustered computing is the practice of pooling the resources of multiple machines and managing their collective capabilities to complete tasks. Computer clusters require a cluster management layer which handles communication between the individual nodes and coordinates work assignment.
- **Data Lake**: Data Lake is a term for a large repository of collected data in a relatively raw state. This is frequently used to refer to the data collected in a big data system which might be unstructured and frequently changing. This differs in spirit to data warehouses (defined below).
- **Data mining**: Data mining is a broad term for the practice of trying to find patterns in large sets of data. It is the process of trying to refine a mass of data into a more understandable and cohesive set of information.
- **Data warehouse**: Data warehouses are large, ordered repositories of data that can be used for analysis and reporting. In contrast to a *data lake*, a data warehouse is composed of data that has been cleaned, integrated with other sources, and is generally well-ordered. Data warehouses are often spoken about in relation to big data, but typically are components of more conventional systems.
- **ETL**: ETL stands for extract, transform, and load. It refers to the process of taking raw data and preparing it for the system's

use. This is traditionally a process associated with data warehouses, but characteristics of this process are also found in the ingestion pipelines of big data systems.

- **Hadoop**: Hadoop is an Apache project that was the early open-source success in big data. It consists of a distributed filesystem called HDFS, with a cluster management and resource scheduler on top called YARN (Yet another Resource Negotiator). Batch processing capabilities are provided by the MapReduce computation engine. Other computational and analysis systems can be run alongside MapReduce in modern Hadoop deployments.
- **In-memory computing**: In-memory computing is a strategy that involves moving the working datasets entirely within a cluster's collective memory. Intermediate calculations are not written to disk and are instead held in memory. This gives in-memory computing systems like Apache Spark a huge advantage in speed over I/O bound systems like Hadoop's MapReduce [16, 18].
- **Machine learning**: Machine learning is the study and practice of designing systems that can learn, adjust, and improve based on the data fed to them. This typically involves implementation of predictive and statistical algorithms that can continually zero in on "correct" behaviour and insights as more data flows through the system [20].
- **Map reduce (big data algorithm)**: Map reduce (the big data algorithm, not Hadoop's MapReduce computation engine) is an algorithm for scheduling work on a computing cluster. The process involves splitting the problem set up (mapping it to different nodes) and computing over them to produce intermediate results, shuffling the results to align like sets, and then reducing the results by outputting a single value for each set.
- **NoSQL**: NoSQL is a broad term referring to databases designed outside of the traditional relational model. NoSQL databases have different trade-offs compared to relational databases, but are often well-suited for big data systems due to their flexibility and frequent distributed-first architecture.

**Stream processing**: Stream processing is the practice of computing over individual data items as they move through a system. This allows for real-time analysis of the data being fed to the system and is useful for time-sensitive operations using high velocity metrics

## VII. CONCLUSION

Big data is a broad, rapidly evolving topic. While it is not well-suited for all types of computing, many organizations are turning to big data for certain types of workloads and using it to supplement their existing analysis and business tools. Big data systems are uniquely suited for surfacing difficult-to-detect patterns and providing insight into behaviour's that are impossible to find through conventional means. By correctly implement systems that deal with big data, organizations can gain incredible value from data that is already available.

## REFERENCES


1. M. K.Kakhani, S. Kakhani and S. R.Biradar, Research issues in bigdata analytics, International Journal of Application or Innovation inEngineering & Management, 2(8) (2015), pp.228-232.
2. A. Gandomi and M. Haider, Beyond the hype: Big data concepts, methods,and analytics, International Journal of Information Management,35(2) (2015), pp.137-144.
3. C. Lynch, Big data: How do your data grow, Nature, 455 (2008),pp.28-29.
4. X. Jin, B. W.Wah, X. Cheng and Y. Wang, Significance and challengesof big data research, Big Data Research, 2(2) (2015), pp.59-64.
5. R. Kitchin, Big Data, new epistemologies and paradigm shifts, BigData Society, 1(1) (2014), pp.1-12.
6. C. L. Philip, Q. Chen and C. Y. Zhang, Data-intensive applications,challenges, techniques and technologies: A survey on big data, InformationSciences, 275 (2014), pp.314-347.
7. K. Kambatla, G. Kollias, V. Kumar and A. Gram, Trends in big dataanalytics, Journal of Parallel and Distributed Computing, 74(7) (2014),pp.2561-2573.
8. S. Del. Rio, V. Lopez, J. M. Bentez and F. Herrera, On the use ofmapreduce for imbalanced big data using random forest, InformationSciences, 285 (2014), pp.112-137.