Understanding Big Data

Manvi Gupta* Roll No. B17092

***Abstract*—Data is growing at a fast pace & using it in an appropriate way is what needs to be learnt. Big data has increased velocity, complexity and variety. What qualifies as big data today may not, tomorrow. The definition will change over time as technology advances! As we go in-depth, we also realize many challenges with big data. An emerging issue with big data is that of privacy. It rises peoples' concern when it comes to privacy! The most exciting thing about big data is what it will do for business when combined with other data. The primary objective of this paper is to highlight some of the important aspects of Big Data, it's management via distributed systems and the future of this technology. We discuss in detail about Hadoop framework and what makes it better than it's other counterparts. Big data will continue to evolve. What we think is big and interesting today might not have any value in a decade, but the generation of new data won't stop, we will still use social media to upload pictures and share stories, we will still need the real-time data for predicting weather conditions & satellite positions!**

***Index Terms*—Big Data, scalability, Hadoop, HDFS, NameNode, DataNode, JobTracker, Blockchain, Cloud Computing**

I. Introduction

As technology continues to grow, so is the amount of data being produced each day. But irrespective of the size, data continues to be a precious & irreplaceable asset. Probably the new and powerful data sources will have the largest impact on advanced analytics in the coming years. The digital data we have can be structured, unstructured, or semi-structured. Before the 1970s, it was the era of mainframes, when the data used to be primitive and structured [6]. Today, unstructured data or the data which is not having a well-defined set of rules like Facebook & Twitter feeds, images, weblogs, etc, comprises of 80% of the total data and has contributed tremendously to the rise of, what we call, Big Data. We need data systems to answer questions based on the information that was acquired in the past upto the present. They combine bits & pieces together to produce their answers [7].

II. Understanding of Big Data

Though there is no clean-cut definition for it, the term Big Data, as the name suggests, may be referred to a data set that is so large in size, that it is beyond human & technical infrastructure to support it's storage & processing. It is used for Analysis and Pattern Realization looking at how the trends in values are changing. It is generally related to human behavioral aspects and human-machine interaction. There's also an explanation that says that neither the "big" part nor the "data" part is the most important aspect when talking about big data [8]. What is most important is what organisations do with big data and that adds value to its existence. Big data is also defined in terms of 5 Vs, i.e. *high-Volume, high-Velocity, high-Value, high-Variety and high-Veracity*.

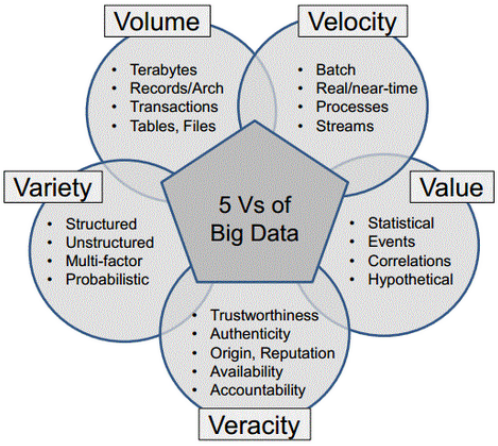

Fig. 1: Five V's of Big Data [5].

III. Challenges Associated with Big Data

Big Data does come with risks. Data is growing at an enormous rate. In the 21st century, almost every big company has to deal with big data challenges at some point of time [2]. One of them being that the organisation will be so overwhelmed with data that it won't progress at all. Most of the data that we have today has been generated in the past couple of years. What needs to be thought about is, "What are we gonna do with so much data?" & "Can it be useful to us for analysis?" [6]. Here are some of the major technical & semantic challenges we face while handling Big Data.

A. Technical Challenges

Though we have come too far as far as computational power and speed is concerned, there still exist some big technical challenges when it comes to data handling. One major such challenge is designing such technologies and hardware which can process the huge volumes of ever growing data. As suggested by Figure 2, the increase in data size has surpassed the capabilities of computation [2]. Transferring data of such large size is also a challenge in itself. *Cloud computing* is one tool to manage infrastructure for big data in terms of cost-efficiency, elasticity, and easy upgrade/downgrade requirements. Organisations need right people attacking big data in an attempt to solve the right problems [8].

Another issue is that costs escalate too fast as too much big data is already gathered before an organisation can decide what to do with it. What we can do is that capture only the samples of new data sources to learn more about them before we actually put them to use [8].

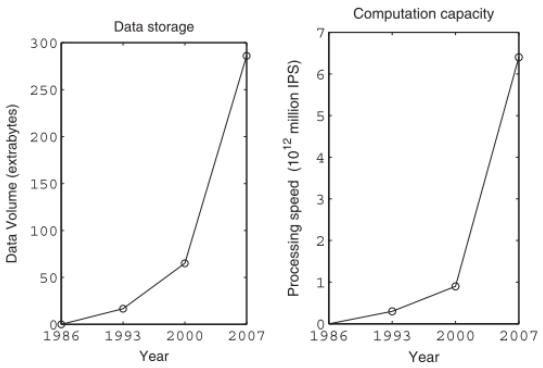

Fig. 2: Data deluge [2].

B. Challenges with Data Semantics

Converting big data into valuable information is a tricky process. The data we have is raw without any specified format, which needs high level of processing and extraction before it can be used for analysis. Another problem that arises, is that of synchronisation, the data we have comes from varied sources and at varied rates. It no more remains synchronized and is difficult to merge.

Similar pieces of data come from varied sources, and these pieces do not always seem to agree. Those pieces are to be matched and integrated for analysis. This data governance or validation of data is complex and often requires a group of people to work together. Another concern that comes is to decide how long the data should be retained. When does a data set become obsolete or irrelevant? [6]. The answer to these questions purely depends on the context and the usefulness of the data for the bigger half of the population. Filtering big data effectively is important, otherwise, there is no meaning of calling it big data when we cannot associate any value with it. The complexity of the rules and magnitude of data being removed or kept at each stage varies with each data source.

Security is also a big concern when it comes to big data. Big data analysts mainly focus on data handling and processing it to find useful results, putting security at later or final stages, which can lead to bigger troubles [9]. Further, we shall see how big data systems are designed in a way that limits the loss of data.

IV. Distributed Data Systems

The data we work upon is accumulated from various different sources, located in the same or different geographies. This data then goes through integration, cleaning up, transformation, and standardization through the process of **Extraction, Transformation & Loading** (ETL), before analysis [6]. Distributed systems have an advantage when it comes to handling big data in terms of storage and accessibility. Most of the data sets that need to be worked up on, are huge in volume and it is almost impossible to manage them using a single system.

A. Desired Properties of a Big Data System

Data systems should not just memorize and regurgitate information [7]. Some information is derived from other pieces of information. When you keep tracking back where the information is derived from, you eventually end up at information that is not derived from anything, or in other words, *raw information*. Such information is what we refer to as data [7]. And, the processing which turns this data into meaningful information after following those many steps is done by complex data systems. Some desired properties of such a system include:

1) Robustness & Fault tolerance: The complex semantics of distributed databases concurrency control mechanisms, make it difficult to predict the reason behind a particular behavior of the system. Robustness involves avoiding such complexities. Another requirement of a system should be that it should be tolerant of *human-errors*, such as deploying incorrect code that corrupts databases [7]. A proper recovery mechanism is what we are referring to here.

2) Scalability: The system should not collapse in case a huge stream of data floods at once. It should be able to maintain it's performance.

3) Ad-hoc Queries: This is an important feature as every large data set as unanticipated values stored within it [7]. Being able to mine a data set arbitrarily is a must for business optimization and working on new applications.

4) Minimal Maintenance: Maintenance, here, refers to the work required to keep a system running smoothly. This includes anticipating when to add machines to scale, keeping processes awake and running, and being able to debug anything that goes wrong in production.

5) Ability to Debug: A big data system must provide all the required information to debug the system as and when things go wrong. Key is to be able to trace, for each value in the system, exactly what caused it to have that value.

B. HADOOP

Hadoop is an open-source computing framework developed by Apache Software Foundation, programmed in Java, which helps us to run applications in a distributed manner across clusters of computers. Primarily, it is meant for the applications that process data, that is huge in terms of volume, variety and value, and cannot be processed by a single system.

Some features that make Hadoop standout among the rest are as follows:

- **Low Cost:** The first and the foremost benefit of using Hadoop is that it is open-source & uses relatively inexpensive hardware.
- **Inherent data protection:** Hadoop protects data & executing applications against hardware failure. If a node fails, the task manager automatically redirects and assigns the task to some other node.
- **Storage flexibility:** Unlike traditional RDBMS, the data can be stored in Hadoop without pre-processing. It can aslo store images, videos, etc.

Though, having listed so many plus points, Hadoop fails when it comes to speed. It has a very low processing speed as compared to the latest frameworks which work at almost 100x.

Hadoop	Spark
Fast	100x Faster
Batch Processing	Real-Time Processing
Stored data on disk	Stores data in memory
Written in Java	Written in Scala

Table 1: Comparison between Hadoop & Spark frameworks

C. Building Blocks of HADOOP

After installation on a fully configured cluster, Hadoop is started by running a set of pre-defined programs on different servers within the network [10]. These can be described as follows:

- **NameNode:** For distributed storage and computation, master/slave architecture is followed by Hadoop, wherein, the distributed storage system is called the *Hadoop Distributed File System* or simply HDFS. The NameNode acts like a bookkeeper for HDFS [10]. It keeps track of the following:
  1) The way files are divided into file blocks
  2) Which nodes store which blocks of files, and
  3) The overall functioning of the HDFS

  Being the most important of all, there's a drawback associated with the NameNode. Its failure leads to the shut down of the entire Hadoop cluster. In case any of the other daemons fails, the cluster still continues to run.

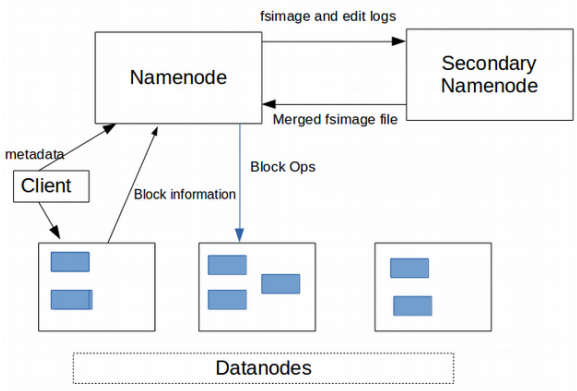

Fig. 3: Functioning of host nodes [4].

- **DataNode:** DataNode reads and writes HDFS blocks to actual files on the local filesystem. This is done by breaking the file into smaller blocks, whose record is kept by the NameNode [10]. Fig. 3 correctly illustrates the role of NameNode & DataNode. DataNodes constantly report to the NameNode about the data they are storing and the changes(create, move, delete) being made to it locally.

- **Secondary NameNode:** The Secondary NameNode (SSN) acts like an assistant NameNode (but cannot completely replace the NameNode) to monitor the HDFS. Each cluster is assigned one SSN. It's main function is to checkpoint the file system metadata of NameNode. It ensures data security as if the NameNode fails, all the files on HDFS are lost, thus SSN sustains a copy of *FsImage file & edits log file*.

- **JobTracker:** It connects Hadoop to the actual application. It acts as a master for task nodes. Whenever the user runs a code, it is the JobTracker which decides which functions to be called & which files to be processed. It also assigns separate TaskTrackers to all the tasks and monitors the code execution. If in case, any of the task fails, it relaunches it automatically [10]. Each Hadoop cluster has one JobTracker.

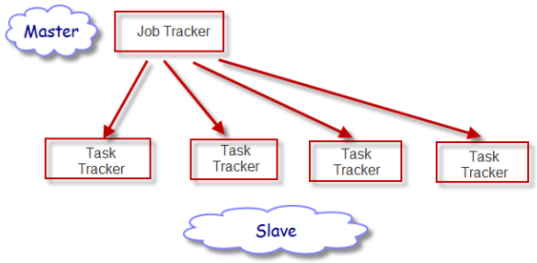

Fig. 4: JobTracker & TaskTrackers in Hadoop [12].

- **TaskTracker:** TaskTrackers manage the tasks assigned to them by the master node (JobTracker) individually. They also have to communicate with the JobTracker that the task is running correctly, else the JobTracker assumes that the task has failed and assigns it to some other node.

D. Working with files in HDFS

To work under MapReduce or alike frameworks, we need filesystems which are designed for large-scale distributed data processing. HDFS is designed to handle such data. It can store a 100 TB data set as a single file!

The workflow goes something like this: When a file of, say, 200 MB comes as an input, first of all, it is split into blocks, by the NameNode [11]. These blocks are then placed into DataNodes but the client doesn't have any knowledge of which DataNodes are free, so the client has to ask the NameNode and then assign the blocks. Now, file replication takes place, and the DataNode keeps sending the status to the NameNode constantly. The DataNode sends an acknowledgement to the client if the process has been completed successfully.

Next step is Data Processing. To run the code on DataNodes, we need the JobTracker. But JobTracker itself doesn't know about the DataNodes that are in use, thus it asks the NameNode & assigns the tasks to TaskNodes which constantly update the JobTracker about the task progress and status. And this way, once the execution completes, we reach to the final results.

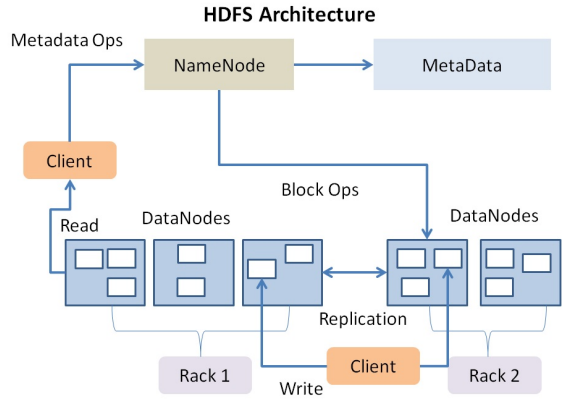

Fig. 5: HDFS [16].

V. MapReduce: A Paradigm for Big Data Computing

MapReduce is a distributed computing model that was originally pioneered by Google and provides primitives for scalable & fault tolerant batch computation [13]. Computations are written in terms of *map* and *reduce* functions that manipulate key-value pairs. They are flexible enough to implement any function. That function then runs over the massive data input in a distributed and robust manner. It lets you focus more on *what* task needs to be done, instead of *how* it's done [13]. Figure 6 describes the working of the algorithm.

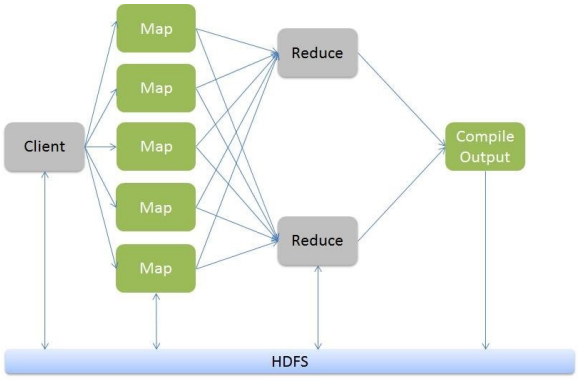

Fig. 6: MapReduce work flow [14].

A. Low-level nature of MapReduce

Manually written MapReduce programs are long and difficult to understand [13]. Running a MapReduce job requires more than just a mapper and a reducer, it also needs to know where to read its input and where to write its output.

Another hardship comes when there arises a need to implement joins in MapReduce. To do a join in MapReduce, we need to read 2 independent datasets in a single MapReduce job, and thus the job should be able to distinguish between records from the 2 datasets [13].

B. Pipe Diagrams

All big data computational problems can be represented in the form of pipe diagrams. Pipe diagrams aren't a hypothetical concept. The idea behind pipe diagrams is to think of processing in terms of tuples, functions, filters, aggregators, joins, and merges. Every pipe diagram can be translated to MapReduce [13]. *Functions and filters* look at 1 record at a time so they can be a part of map step or a reduce step. *Group by* is easily translated by the key emitted in the map step. *Aggregation* happens in the reduce step [13].

VI. Hadoop Vs. Cloud Vs. Blockchain

A. Cloud Computing

Cloud Computing mainly aims at improving the data storage techniques & power of Data. It includes the data centers accessible to a wide range of users over the internet and is not just limited to a specific hard drive. There are many good and bad points, or strengths and weaknesses associated with cloud computing. an organisation needs to know all the pros and cons beforehand to make a learned choice.

B. Comparison between Hadoop and Cloud Computing:

- In cloud computing, all the data, applications and softwares are stored on the cloud server, & can be accessed through the internet. In Hadoop, all files are stored & processed in HDFS via all the Nodes across clusters.
- Sometimes, Cloud storage has greater request latency as compared to HDFS.
- Cloud provides a wide availability of data as it can be accessed from anywhere. There is no risk of NameNode failure or the entire cluster failure as in Hadoop.
- Cloud is cost-efficient as there is no need for buying too much hardware and you can pay as per the storage requirements.
- Cloud MapReduce does not provide its own implementation like Hadoop, instead, it relies on the cloud service provider's infrastructure.

C. Comparison between Hadoop and Blockchain:

Hadoop and Blockchain are two completely different concepts, in terms of purpose as well as functioning. Blockchain stores the information in terms of blocks, which once written are impossible to modify, unlike Hadoop wihch provides higher reliability.

VII. Big Data in Real Life

These days, the word "Big Data" is trending a lot, especially among big companies who are ready to take huge risks relying on this new technology. One of the cases includes Google's flu trends, which was a big failure due to the uncontrollably large volume of data [1]. Here we shall discuss another *"Big Application of Big Data"*, by China.

A. China's "Trustworthiness"

China is planning, by 2020, to provide every citizen a rating in terms of *trustworthiness* which shall include everything starting from friend suggestions to clothing habits [3]. At first, it seems like a quintessential tool, which can prove to be beneficial for many people. But although surveillance technology is appreciated by a few, the majority population has more privacy concerns. This Big Data application is very likely to make the nation less repressive [3].

This technology would rely on the huge volumes of data collected from wide ranging sources which may or may not be relied upon. Then comes the issue of data synchronization as the data will be in a variety of different formats which are hard to process together.

There has also been a rise in other Big Data technologies in China, the reason being China has comparatively lesser privacy awareness than other nations. This is reflected through many platforms, such as *WeChat* allows us to make numerous friends without knowing anything about them. On a similar note, *Alibaba* allows trade between completely unknown group of people [3].

In 2016, China also introduced a Big Data inspired, traffic management system called the *City Brain* [3]. What makes it different from Google maps is that City Brain is in collaboration with the city Government, and along with from traffic management, it can also provide video footage of any kind of incidents on roads.

VIII. Future of Big Data Technology

Today's Big Data is not tomorrow's Big Data. [8].

As mentioned in the beginning, there is no concrete definition for big data, rather it is defined in terms of technologies and resources available to us then. Ans as technology processes, so will the capability of our systems and thus the definition for "BIG". Transactional data in the retail, telecommunication, & banking industries were big and hard to handle even a decade ago [8]. And the data was not available widely for analytical purposes or reporting. Today, such data is considered as a necessary and fundamental asset. Similarly, what we are intimidated by today won't be so scary a few years down the lane.

Though having encountered many failures, the future of Big Data can be quite surprising. Some quite imaginative ideas are as follows:

- What if along with web browsing history, browsers start keeping track of our mouse or cursor movements too!
- If all the items in all the stores had RFID tags and you just have to go and pick the stuff back home! [8]. Also, the small chips packet capturing information about temperature, humidity, speed, etc. Sounds crazy but something that needs to be focused upon is that, this will lead to a big boom in the size of data, it will be unthinkably huge.
- Imagine your car warning you each time you take a wrong turn or a wrong choose the wrong way.

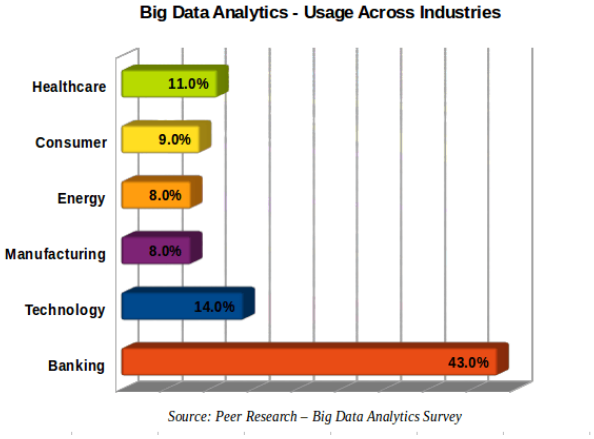

Fig. 7: Usage of Big Data [15].

All of these technologies might seem impossible or some might seem achievable but there's one thing in common, i.e. all of them can function only if with the help of massive datasets collected from real life experiences.

IX. Conclusion

Some important characteristics make big data different from traditional data sources. It is not simply an extended collection of data. Big data is just the next wave of new, bigger data that pushes current limits. Many sources of big data are actually semi-structured or multi-structured, not unstructured. Such data does have a logical flow to it that can be understood so that information can be extracted from it for analysis. We have made it possible, to a great extent, to process and analyze big data but who knows its rate of growth ten years from now. It can be exciting and at the same time, challenging to study how this transition occurs and how we can benefit from it.

References

[1] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani, "The Parable of Google Flu: Traps in Big Data Analysis," *February 2013*

[2] C.L. Philip Chen, and Chun-Yang Zhang, 'Data-intensive applications, challenges, techniques and technologies: A survey on Big Data'

[3] MIT Technology Review, "China's use of big data might actually make it less big brother-ish", August 2018

[4] Functioning of host nodes, https://bit.ly/2mhH1Gd

[5] Five V's of Big Data, https://bit.ly/2klz8yQ

[6] Seema Acharya, and Subhashini Chellappan, "Introduction to Big Data" in *Big Data and Analytics*, Ch. 2, Wiley India, 2015

[7] Nathan Marz, James Warren, "A new paradigm for Big Data", in *Big Data*, Ch. 1, Manning Publications USA, 2015 Edition

[8] Bill Franks, "What is Big Data and Why Does it Matter?", in *Taming the Big Data Tidal Wave*, Hoboken, NJ: Wiley, 2015 Edition

[9] Alex Bekker, "The Scary Seven: big data challenges and ways to solve them", https://bit.ly/2mi9HyR, March 2018

[10] Chuck Lam, "Starting Hadoop" in *Hadoop in Action*, Ch. 2, Manning Publications USA, 2015 Edition

[11] HDFS File Processing Working of HDFS, https://bit.ly/2lO9BPa

[12] JobTracker and TaskTracker, https://bit.ly/2mcxQ9K

[13] Nathan Marz, James Warren, "Batch Layer", in *Big Data*, Ch. 6, Manning Publications USA, 2015 Edition

[14] Hadoop MapReduce Tutorial, https://bit.ly/2ma2AYX

[15] What are the Big Data Hadoop Challenges?, https://bit.ly/2ma3vsn, February 2017

[16] HDFS, https://www.whizlabs.com/blog/hdfs-interview-questions/