A

Technical Research Paper

On

# “Big Data & Hadoop”

Submitted By

Sudarshan Sharad Gawale

Submitted To

Techno Mania 2020 State Level Technical Paper
Presentation Competition 2019 – 2020

# Abstract

Everyday enormous amount of data is being produced worldwide. Big Data analytics has brought a big opportunity for organizations. Companies capture trillions of bytes of information about their customers, suppliers, and operations. IT organizations are exploring the analytics technologies to explore web-based data sources and extract value from the social networking boom. In the western world, organizations are wondering about the kind of business intelligence they could derive from all the information they have at their disposal.? The organizations are trying to leverage Big Data by trying to make sense from the data that they have and by securing it. In the next three to five years, there will be a widening gap between companies that understand and exploit Big Data and companies that are aware of it but do not know what to do with it. Already the forward thinking players of the banking, insurance, manufacturing, retail, wholesale, healthcare, communications, transportation, construction, utilities, and education are successfully using big data by exploiting meaningful information from all the Data they have and using those information in formulating their strategic moves. Those companies who will be able to use Big Data successfully will be clearly ahead of those who will react slowly to capitalize on Big Data.

# Index

# 1. What is Big Data?


**Fig 1.1 :** What is Big Data ?

There is no place where Big Data does not exist! The curiosity about what is Big Data has been soaring in the past few years. Let me tell you some mind-boggling facts! Forbes reports that every minute, users watch *4.15 million YouTube videos*, send *456,000 tweets* on Twitter, post *46,740 photos* on Instagram and there are *510,000 comments* posted and *293,000 statuses* updated on Facebook!

Just imagine the huge chunk of data that is produced with such activities. This constant creation of data using social media, business applications, telecom and various other domains is leading to the formation of Big Data.

In order to explain **what is Big Data**, I will be covering the following topics:

- Evolution of Big Data
- Big Data Defined
- Characteristics of Big Data
- Big Data Analytics
- Industrial Applications of Big Data
- Scope of Big Data

## 1.1 Evolution of Big Data

Before exploring what is Big Data, let me begin by giving some insight into why the term Big Data has gained so much importance.

When was the last time you guys remember using a floppy or a CD to store your data? Let me guess, had to go way back in the early 21st century right? The use of manual paper records, files, floppy and discs have now become obsolete. The reason for this is the exponential growth of data. People began storing their data in relational database systems but with the hunger for new inventions, technologies, applications with quick response time and with the introduction of the internet, even that is insufficient now. This generation of continuous and massive data can be referred to as Big Data. There are a few other factors that characterize Big Data which I will be explaining later in this blog.

Forbes reports that there are 2.5 quintillion bytes of data created each day at our current pace, but that pace is only accelerating. Internet of Things(IoT) is one such technology which plays a major role in this acceleration. 90% of all data today was generated in the last two years

## 1.2 What is Big Data?

So before I explain what is Big Data, let me also tell you what it is not! The most common myth associated with Big Data is that it is just about the size or volume of data. But actually, it's not just about the "big" amounts of data being collected. **Big Data** refers to the large amounts of data which is pouring in from various data sources and has different formats. Even previously there was huge data which were being stored in databases, but because of the varied nature of this Data, the traditional relational database systems are incapable of handling this Data. Big Data is much more than a collection of datasets with different formats, it is an important asset which can be used to obtain enumerable benefits.

**The three different formats of big data are:**

1. *Structured:* Organised data format with a fixed schema. Ex: RDBMS
2. *Semi-Structured:* Partially organised data which does not have a fixed format. Ex: XML, JSON
3. *Unstructured:* Unorganised data with an unknown schema. Ex: Audio, video files etc.

# 2. Characteristics of Big Data

These are the following characteristics associated with Big Data:

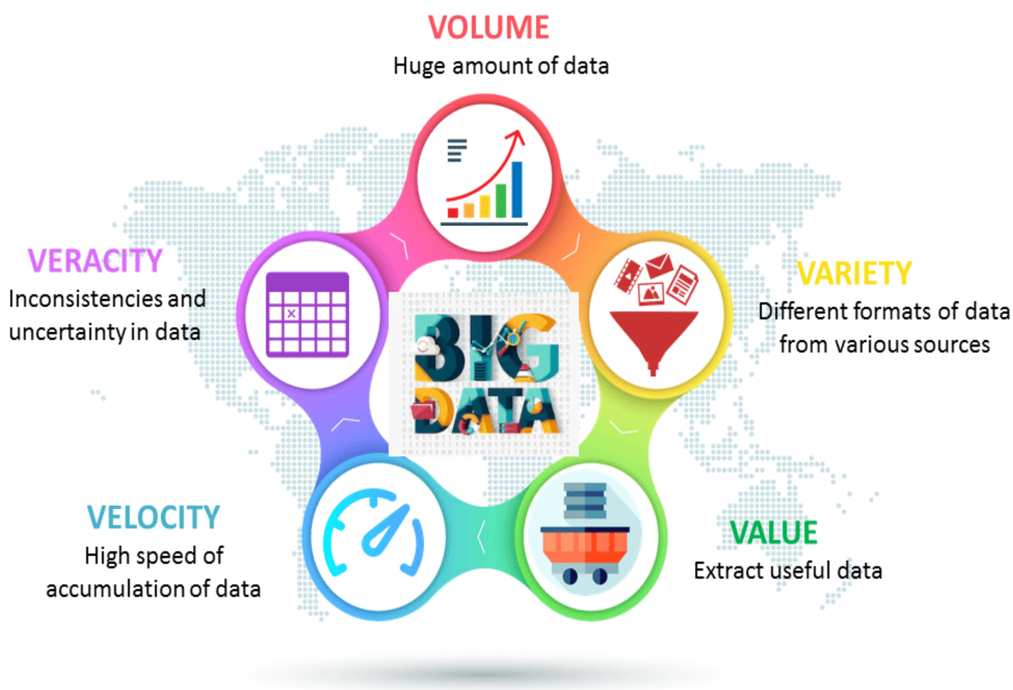


**Fig. 2.1 :** Big Data Characteristics

The above image depicts the five V's of Big Data but as and when the data keeps evolving so will the V's. I am listing five more V's which have developed gradually over time:

- Validity: correctness of data
- Variability: dynamic behaviour
- Volatility: tendency to change in time
- Vulnerability: vulnerable to breach or attacks
- Visualization: visualizing meaningful usage of data

## 2.1 Big Data Analytics

Now that I have told you what is Big Data and how it's being generated exponentially, let me present to you a very interesting example of how *Starbucks*, one of the leading coffeehouse chain is making use of this Big Data.

I came across this article by Forbes which reported how *Starbucks* made use of Big Data to analyse the preferences of their customers to enhance and personalize their experience. They analysed their member's coffee buying habits along with their preferred drinks to what time of day they are usually ordering. So, even when people visit a "new" Starbucks location, that store's point-of-sale system is able to identify the customer through their smartphone and give the barista their preferred order. In addition, based on ordering preferences, their app will suggest new products that the customers might be interested in trying. This my friends is what we call Big Data Analytics.

## 2.2 Big Data Training

Basically, Big Data Analytics is largely used by companies to facilitate their growth and development. This majorly involves applying various data mining algorithms on the given set of data, which will then aid them in better decision making.

There are multiple tools for processing Big Data such as **Hadoop**, **Pig**, **Hive**, **Cassandra**, **Spark**, **Kafka**, etc. depending upon the requirement of the organisation.


**Fig. 2.2 :** Big Data Tools

## 2.3 Big Data Applications

These are some of the following domains where **Big Data Applications** has been revolutionized:

- **Entertainment:** Netflix and Amazon use Big Data to make shows and movie recommendations to their users.
- **Insurance:** Uses Big data to predict illness, accidents and price their products accordingly.
- **Driver-less Cars:** Google's driver-less cars collect about one gigabyte of data per second. These experiments require more and more data for their successful execution.
- **Education:** Opting for big data powered technology as a learning tool instead of traditional lecture methods, which enhanced the learning of students as well aided the teacher to track their performance better.
- **Automobile:** Rolls Royce has embraced Big Data by fitting hundreds of sensors into its engines and propulsion systems, which record every tiny detail about their operation. The changes in data in real-time are reported to engineers who will decide the best course of action such as scheduling maintenance or dispatching engineering teams should the problem require it.
- **Government:** A very interesting use of Big Data is in the field of politics to analyse patterns and influence election results. Cambridge Analytica Ltd. is one such organisation which completely drives on data to change audience behaviour and plays a major role in the electoral process.

# 3. What is Hadoop?

Big Data is emerging as an opportunity for organizations. Now, organizations have realized that they are getting lots of benefits by Big Data Analytics, as you can see in the below image. They are examining large data sets to uncover all hidden patterns, unknown correlations, market trends, customer preferences and other useful business information.

These analytical findings are helping organizations in more effective marketing, new revenue opportunities, better customer service. They are improving operational efficiency, competitive advantages over rival organizations and other business benefits.

So, let us move ahead and know the problems associated with traditional approach in en-cashing Big data opportunities.

## 3.1 Problems with Traditional Approach

In traditional approach, the main issue was handling the heterogeneity of data i.e. structured, semi-structured and unstructured. The RDBMS focuses mostly on structured data like banking transaction, operational data etc. and Hadoop specializes in semi-structured, unstructured data like text, videos, audios, Facebook posts, logs, etc. RDBMS technology is a proven, highly consistent, matured systems supported by many companies. While on the other hand, Hadoop is in demand due to Big Data, which mostly consists of unstructured data in different formats.

Now let us understand what are the major problems associated with Big Data. So that, moving ahead we can understand how Hadoop emerged as a solution.

### 3.1.1 The first problem is storing the colossal amount of data.

Storing this huge data in a traditional system is not possible. The reason is obvious, the storage will be limited only to one system and the data is increasing at a tremendous rate.

### 3.1.2 Second problem is storing heterogeneous data.

Now, we know storing is a problem, but let me tell you, it is just a part of the problem. Since we discussed that the data is not only huge, but it is present in various formats as well like: Unstructured, Semi-structured and Structured. So, you need to make sure that, you have a system to store all these varieties of data, generated from various sources.

### 3.1.3 Third problem is accessing and processing speed.

The hard disk capacity is increasing but the disk transfer speed or the access speed is not increasing at similar rate. Let me explain you this with an example: If you have only one 100 Mbps I/O channel and you are processing 1TB of data, it will take around 2.91 hours. Now, if you have four machines with one I/O channel, for the same amount of data it will take 43 minutes approx. Thus, accessing and processing speed is the bigger problem than storing Big Data.

Before understanding what is Hadoop, let us first look at the evolution of Hadoop over the period of time.

## 3.2 Evolution of Hadoop

In 2003, Doug Cutting launches project Nutch to handle billions of searches and indexing millions of web pages. Later in Oct 2003 – Google releases papers with GFS (Google File System). In Dec 2004, Google releases papers with MapReduce. In 2005, Nutch used GFS and MapReduce to perform operations. In 2006, Yahoo created Hadoop based on GFS and MapReduce with Doug Cutting and team. You would be surprised if I would tell you that, in 2007 Yahoo started using Hadoop on a 1000 node cluster.

Later in Jan 2008, Yahoo released Hadoop as an open source project to Apache Software Foundation. In Jul 2008, Apache tested a 4000 node cluster with Hadoop successfully. In 2009, Hadoop successfully sorted a petabyte of data in less than 17 hours to handle billions of searches and indexing millions of web pages. Moving ahead in Dec 2011, Apache Hadoop released version 1.0. Later in Aug 2013, Version 2.0.6 was available.

When we were discussing about the problems, we saw that a distributed system can be a solution and Hadoop provides the same. Now, let us understand what is Hadoop.

## 3.3 What is Hadoop?

Hadoop is a framework that allows you to first store Big Data in a distributed environment, so that, you can process it parallely. There are basically two components in Hadoop:

The first one is **HDFS** for storage (Hadoop distributed File System), that allows you to store data of various formats across a cluster. The second one is **YARN**, for resource management in Hadoop. It allows parallel processing over the data, i.e. stored across HDFS.

Let us first understand HDFS.

### 3.3.1 HDFS

HDFS creates an abstraction, let me simplify it for you. Similar as virtualization, you can see HDFS logically as a single unit for storing Big Data, but actually you are storing your data across multiple nodes in a distributed fashion. HDFS follows master-slave architecture.

In HDFS, Namenode is the master node and Datanodes are the slaves. Namenode contains the metadata about the data stored in Data nodes, such as which data block is stored in which data node, where are the replications of the data block kept etc. The actual data is stored in Data Nodes.

I also want to add, we actually replicate the data blocks present in Data Nodes, and the default replication factor is 3. Since we are using commodity hardware and we know the failure rate of these hardwares are pretty high, so if one of the DataNodes fails, HDFS will still have the copy of those lost data blocks. You can also configure replication factor based on your requirements.

### 3.3.2 YARN

YARN performs all your processing activities by allocating resources and scheduling tasks.

It has two major components, i.e. ResourceManager and NodeManager.

ResourceManager is again a master node. It receives the processing requests and then passes the parts of requests to corresponding NodeManagers accordingly, where the actual processing takes place. NodeManagers are installed on every DataNode. It is responsible for the execution of the task on every single DataNode.

- It is a node level component (one on each node) and runs on each slave machine
- It is responsible for managing containers and monitoring resource utilization in each container
- It also keeps track of node health and log management
- It continuously communicates with ResourceManager to remain up-to-date

# 4. Hadoop : Big Data & Hadoop – Restaurant Analogy

## 4.1 Restaurant Analogy

Let us take an analogy of a restaurant to understand the problems associated with Big Data and how Hadoop solved that problem.

Bob is a businessman who has opened a small restaurant. Initially, in his restaurant, he used to receive two orders per hour and he had one chef with one food shelf in his restaurant which was sufficient enough to handle all the orders.

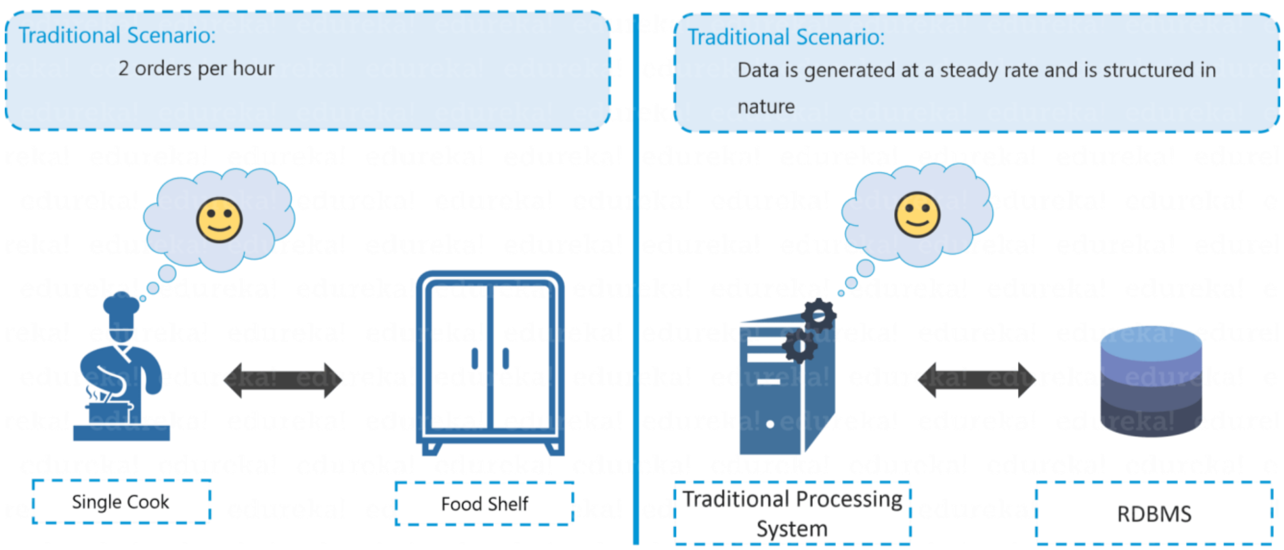


**Fig 4.1 :** Hadoop – Traditional Restaurant Scenario

Now let us compare the restaurant example with the traditional scenario where data was getting generated at a steady rate and our traditional systems like RDBMS is capable enough to handle it, just like Bob's chef. Here, you can relate the data storage with the restaurant's food shelf and the traditional processing unit with the chef as shown in the figure above.

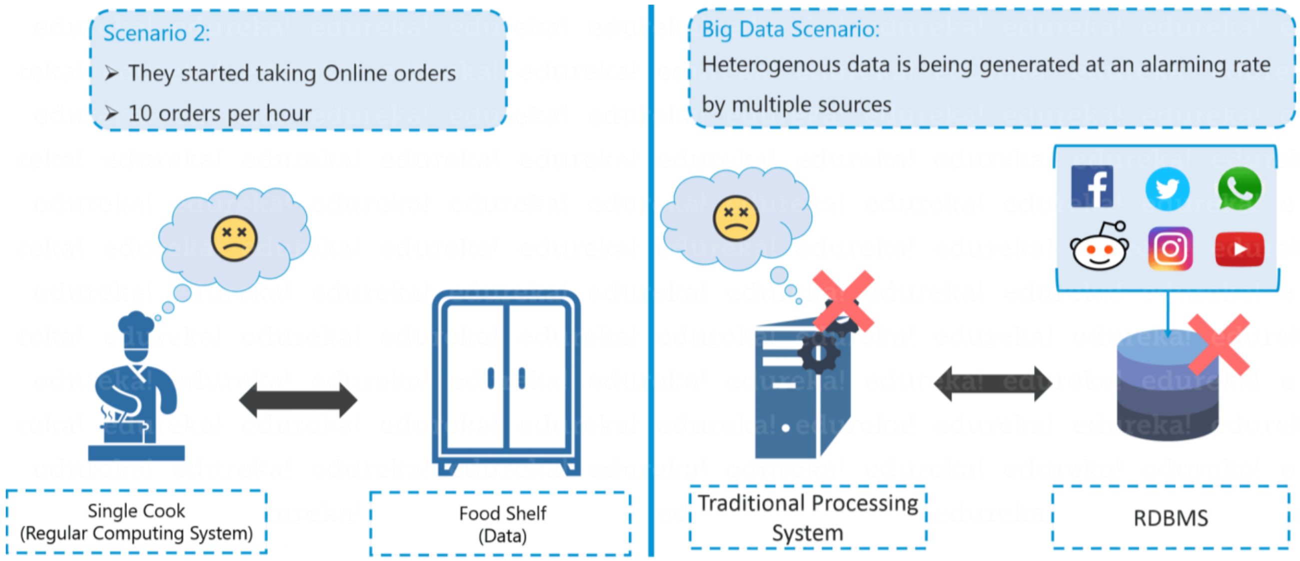


**Fig 4.2 :** Hadoop – Traditional Scenario

After few months, Bob thought of expanding his business and therefore, he started taking online orders and added few more cuisines to the restaurant's menu in order to engage a larger audience. Because of this transition, the rate at which they were receiving orders rose to an alarming figure of 10 orders per hour and it became quite difficult for a single cook to cope up with the current situation. Aware of the situation in processing the orders, Bob started thinking about the solution.

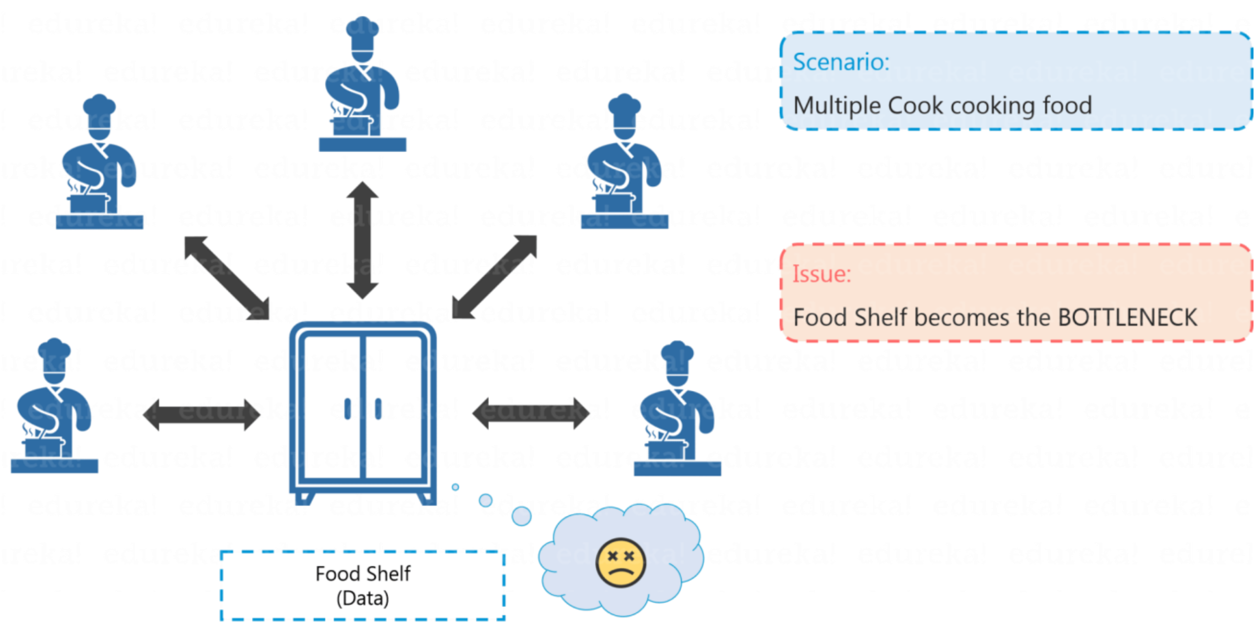


**Fig 4.3 :** Hadoop – Distributed Processing Scenario

Similarly, in Big Data scenario, the data started getting generated at an alarming rate because of the introduction of various data growth drivers such as social media, smartphones etc. Now, the traditional system, just like cook in Bob's restaurant, was not efficient enough to handle this sudden change. Thus, there was a need for a different kind of solutions strategy to cope up with this problem.

After a lot of research, Bob came up with a solution where he hired 4 more chefs to tackle the huge rate of orders being received. Everything was going quite well, but this solution led to one more problem. Since four chefs were sharing the same food shelf, the very food shelf was becoming the bottleneck of the whole process. Hence, the solution was not that efficient as Bob thought.

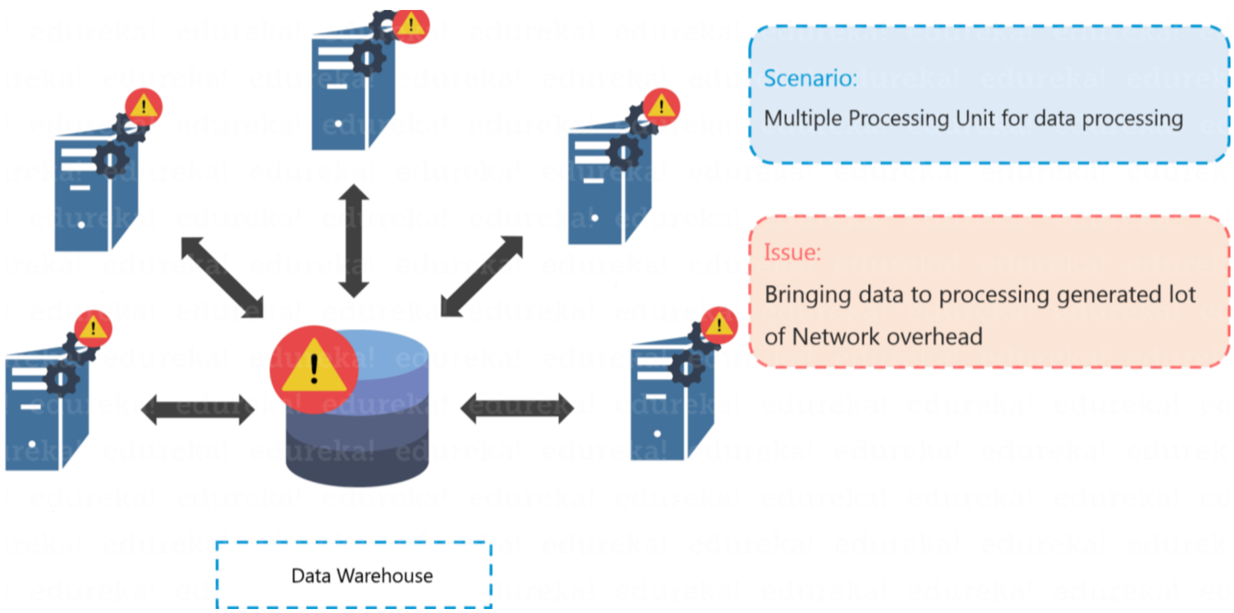


**Fig 4.4 :** Hadoop Tutorial – Distributed Processing Scenario Failure

Similarly, to tackle the problem of processing huge datasets, multiple processing units were installed so as to process the data parallelly (just like Bob hired 4 chefs). But even in this case, bringing multiple processing units was not an effective solution because: the centralized storage unit became the bottleneck. In other words, the performance of the whole system is driven by the performance of the central storage unit. Therefore, the moment our central storage goes down, the whole system gets compromised. Hence, again there was a need to resolve this single point of failure.

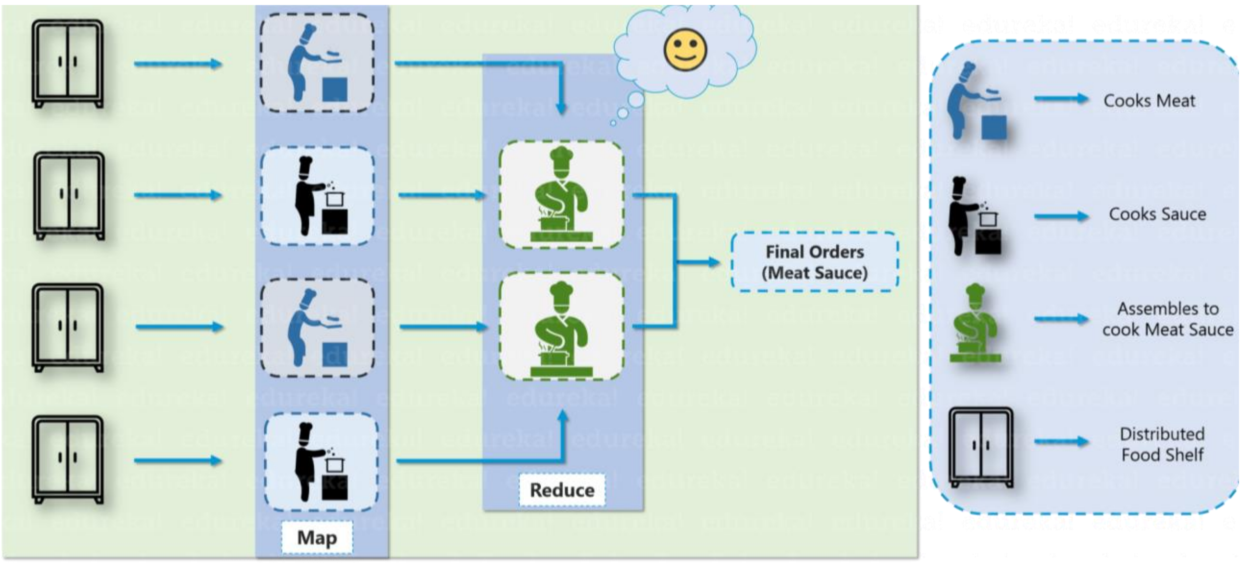


**Fig 4.5 :** Hadoop– Solution to Restaurant Problem

Bob came up with another efficient solution, he divided all the chefs in two hierarchies, i.e. junior and head chef and assigned each junior chef with a food shelf. Let us assume that the dish is Meat Sauce. Now, according to Bob's plan, one junior chef will prepare meat and the other junior chef will prepare the sauce. Moving ahead they will transfer both meat and sauce to the head chef, where the head chef will prepare the meat sauce after combining both the ingredients, which then will be delivered as the final order.

Hadoop functions in a similar fashion as Bob's restaurant. As the food shelf is distributed in Bob's restaurant, similarly, in Hadoop, the data is stored in a distributed fashion with replications, to provide fault tolerance. For parallel processing, first the data is processed by the slaves where it is stored for some intermediate results and then those intermediate results are merged by master node to send the final result.

Now, you must have got an idea why Big Data is a problem statement and how Hadoop solves it. As we just discussed above, there were three major challenges with Big Data:

- **The first problem is storing the colossal amount of data.** Storing huge data in a traditional system is not possible. The reason is obvious, the storage will be limited to one system and the data is increasing at a tremendous rate.
- **The second problem is storing heterogeneous data.** Now we know that storing is a problem, but let me tell you it is just one part of the problem. The data is not only huge, but it is also present in various formats i.e. unstructured, semi-structured and structured. So, you need to make sure that you have a system to store different types of data that is generated from various sources.

- **Finally let's focus on the third problem, which is the processing speed.** Now the time taken to process this huge amount of data is quite high as the data to be processed is too large.

## 4.2 Hadoop as a Solution

Let's understand the how Hadoop provided the solution to the Big Data problems that we just discussed.

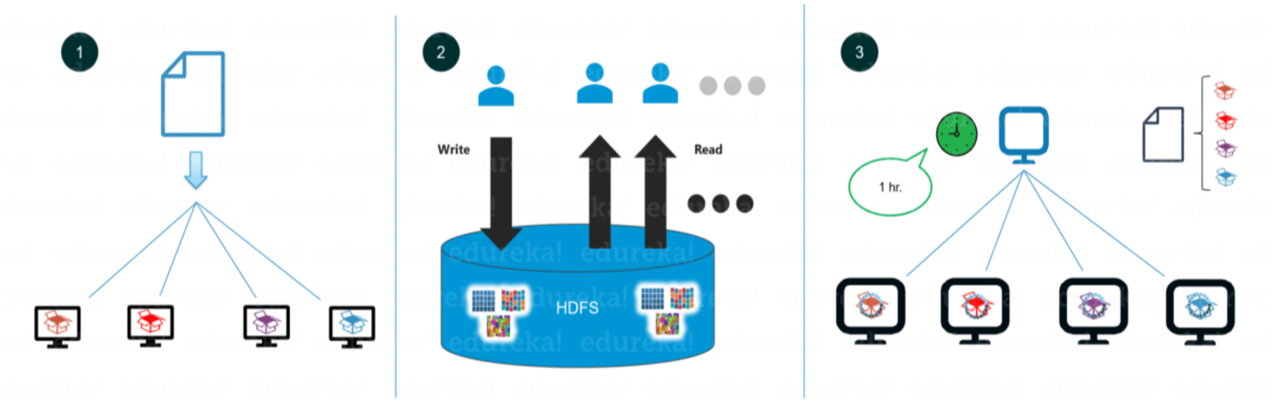


**Fig 4.6 :** What is Hadoop – Hadoop-as-a-Solution

**The first problem is storing Big data**.

HDFS provides a distributed way to store Big data. Your data is stored in blocks across the DataNodes and you can specify the size of blocks. Basically, if you have 512MB of data and you have configured HDFS such that, it will create 128 MB of data blocks. So HDFS will divide data into 4 blocks as 512/128=4 and store it across different DataNodes, it will also replicate the data blocks on different DataNodes. Now, as we are using commodity hardware, hence storing is not a challenge.

It also solves the scaling problem. It focuses on **horizontal scaling** instead of vertical scaling. You can always add some extra data nodes to HDFS cluster as and when required, instead of scaling up the resources of your DataNodes. Let me summarize it for you basically for storing 1 TB of data, you don't need a 1TB system. You can instead do it on multiple 128GB systems or even less.

**Next problem was storing the variety of data**.

With HDFS you can store all kinds of data whether it is structured, semi-structured or unstructured. Since in HDFS, there is *no pre-dumping schema validation*. And it also follows write once and read many model. Due to this, you can just write the data once and you can read it multiple times for finding insights.

**Third challenge was accessing & processing the data faster.**

Yes, this is one of the major challenges with Big Data. In order to solve it, we move processing to data and not data to processing. What does it mean? Instead of moving data to the master node and then processing it. In MapReduce, the processing logic is sent to the various slave nodes & then data is processed parallely across different slave nodes. Then the processed results are sent to the master node where the results is merged and the response is sent back to the client.

In YARN architecture, we have ResourceManager and NodeManager. ResourceManager might or might not be configured on the same machine as NameNode. But, NodeManagers should be configured on the same machine where DataNodes are present.

# 5. Hadoop Features

## 5.1 Features

### Reliability*:*

When machines are working in tandem, if one of the machines fails, another machine will take over the responsibility and work in a reliable and fault tolerant fashion. Hadoop infrastructure has inbuilt fault tolerance features and hence, Hadoop is highly reliable.

### Economical*:*

Hadoop uses commodity hardware (like your PC, laptop). For example, in a small Hadoop cluster, all your DataNodes can have normal configurations like 8-16 GB RAM with 5-10 TB hard disk and Xeon processors, but if I would have used hardware-based RAID with Oracle for the same purpose, I would end up spending 5x times more at least. So, the cost of ownership of a Hadoop-based project is pretty minimized. It is easier to maintain the Hadoop environment and is economical as well. Also, Hadoop is an open source software and hence there is no licensing cost.

### Scalability*:*

Hadoop has the inbuilt capability of integrating seamlessly with cloud-based services. So, if you are installing Hadoop on a cloud, you don't need to worry about the scalability factor because you can go ahead and procure more hardware and expand your setup within minutes whenever required.

### Flexibility*:*

Hadoop is very flexible in terms of ability to deal with all kinds of data. We discussed "Variety, where data can be of any kind and Hadoop can store and process them all, whether it is structured, semi-structured or unstructured data.

These 4 characteristics make Hadoop a front-runner as a solution to Big Data challenges. Now that we know what is Hadoop, we can explore the core components of Hadoop. Let us understand, what are the core components of Hadoop.

## 5.2 Hadoop Core Components

While setting up a Hadoop cluster, you have an option of choosing a lot of services as part of your Hadoop platform, but there are two services which are always mandatory for setting up Hadoop. One is **HDFS (storage)** and the other is **YARN (processing)**. HDFS stands for **Hadoop Distributed File System**, which is a scalable storage unit of Hadoop whereas YARN is used to process the data i.e. stored in the HDFS in a distributed and parallel fashion.

#### 5.2.1 HDFS

Let us go ahead with HDFS first. The main components of **HDFS** are: **NameNode** and **DataNode**. Let us talk about the roles of these two components in detail.

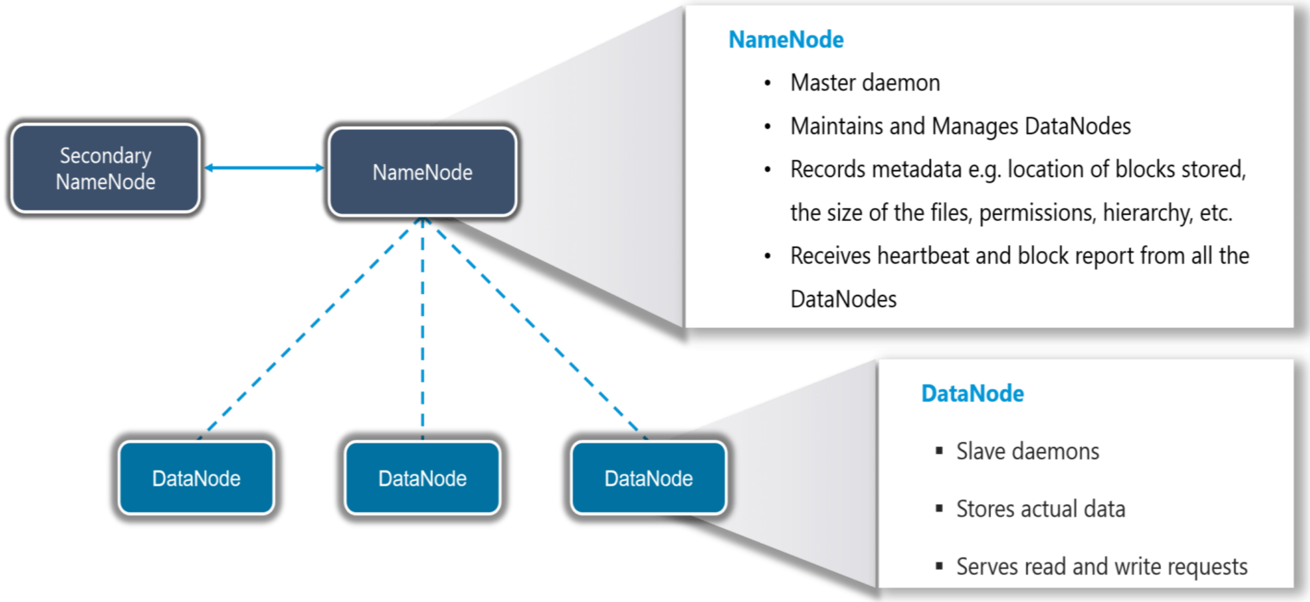


**Fig 5.1 :** Hadoop – HDFS

**NameNode**

- It is the master daemon that maintains and manages the DataNodes (slave nodes)
- It records the metadata of all the blocks stored in the cluster, e.g. location of blocks stored, size of the files, permissions, hierarchy, etc.
- It records each and every change that takes place to the file system metadata
- If a file is deleted in HDFS, the NameNode will immediately record this in the EditLog
- It regularly receives a Heartbeat and a block report from all the DataNodes in the cluster to ensure that the DataNodes are live
- It keeps a record of all the blocks in the HDFS and DataNode in which they are stored
- It has high availability and federation features which I will discuss in **HDFS architecture** in detail

**DataNode**

- It is the slave daemon which run on each slave machine
- The actual data is stored on DataNodes
- It is responsible for serving read and write requests from the clients
- It is also responsible for creating blocks, deleting blocks and replicating the same based on the decisions taken by the NameNode
- It sends heartbeats to the NameNode periodically to report the overall health of HDFS, by default, this frequency is set to 3 seconds

So, this was all about HDFS in nutshell. Now, let move ahead to our second fundamental unit of Hadoop i.e. YARN.

### 5.2.2 YARN

*YARN* comprises of two major component: **ResourceManager** and **NodeManager**.

**ResourceManager**

- It is a cluster level (one for each cluster) component and runs on the master machine
- It manages resources and schedule applications running on top of YARN
- It has two components: Scheduler & ApplicationManager
- The Scheduler is responsible for allocating resources to the various running applications
- The ApplicationManager is responsible for accepting job submissions and negotiating the first container for executing the application
- It keeps a track of the heartbeats from the Node Manager

**NodeManager**

- It is a node level component (one on each node) and runs on each slave machine
- It is responsible for managing containers and monitoring resource utilization in each container
- It also keeps track of node health and log management
- It continuously communicates with ResourceManager to remain up-to-date

## Hadoop Ecosystem

So far you would have figured out that Hadoop is neither a programming language nor a service, it is a platform or framework which solves Big Data problems. You can consider it as a suite which encompasses a number of services for ingesting, storing and analyzing huge data sets along with tools for configuration management.

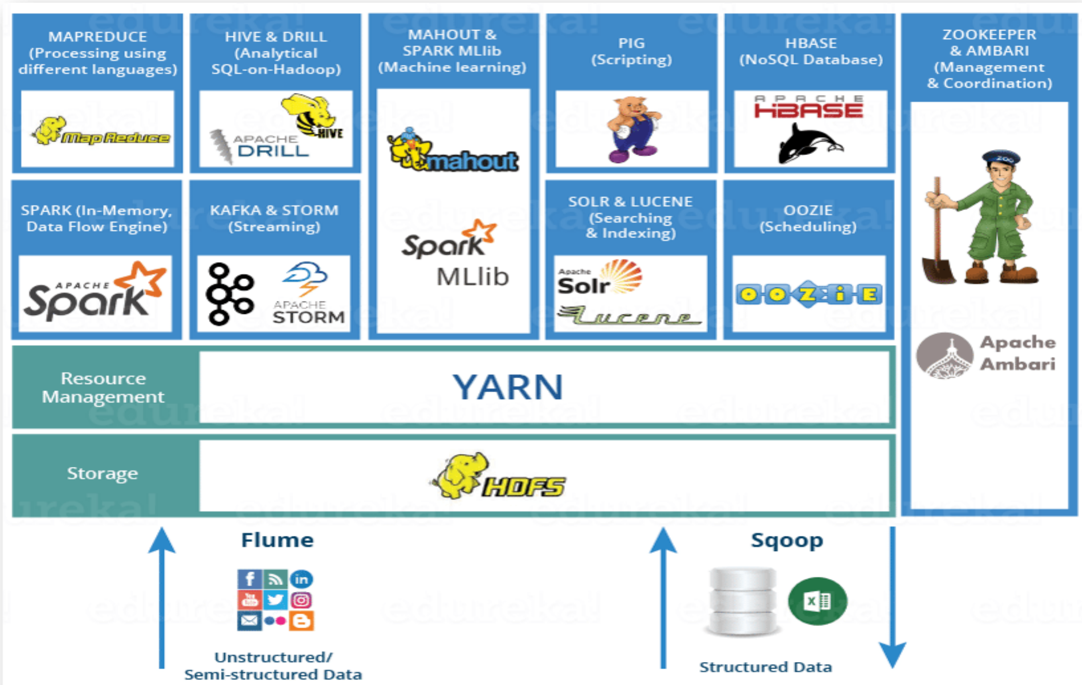


**Fig 5.2 :** Hadoop Ecosystem

# 6. Conclusion :

- Big Data is growing rapidly as never before.
- Big Data is important as organizations can mine meaningful insights from this big data for their business.
- To cater this increasing demand of storage and computation, frameworks like Hadoop are essential.
- Hadoop stores and processes big data efficiently and with faster execution time and also provides robust fault tolerance.

# 7. Image References :

| **Images No** | **Reference** |
|---|---|
| 1.1 | https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwjVtMayr97kAhXA8HMBHX2HBJMQjRx6BAgBEAQ&url=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fwhat-is-big-data%2F&psig=AOvVaw2sOf67uyH9_qZ7TbQ_OzJ-&ust=1569031825757083 |
| 2.1 | https://www.edureka.co/blog/what-is-big-data/ |
| 2.2 | https://www.edureka.co/blog/what-is-big-data/ |
| 4.1 | https://www.edureka.co/blog/hadoop-tutorial/ |
| 4.2 | https://www.edureka.co/blog/hadoop-tutorial/ |
| 4.3 | https://www.edureka.co/blog/hadoop-tutorial/ |
| 4.4 | https://www.edureka.co/blog/hadoop-tutorial/ |
| 4.5 | https://www.edureka.co/blog/hadoop-tutorial/ |
| 4.6 | https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwjIqanFsN7kAhXc7nMBHd0cDa4QjRx6BAgBEAQ&url=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fwhat-is-hadoop%2F&psig=AOvVaw04a_QL8D0lBrIL0T6t8W7Y&ust=1569033799381620 |
| 5.1 | https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2017/05/HDFS-What-is-Hadoop-Edureka.png |
| 5.2 | https://www.edureka.co/blog/hadoop-ecosystem |

# 8. References :


[1]. https://hadoop.apache.org/docs/current/index.html

[2]. https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

[3]. [https://www.sas.com/en_in/insights/big-data/hadoop.html

[4]. https://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.htm

[5]. https://www.edureka.co/blog/hadoop-tutorial/

[6]. https://www.javatpoint.com/hadoop-tutorial

[7]. https://www.oracle.com/big-data/guide/what-is-big-data.html

[8]. https://www.edureka.co/blog/what-is-big-data/

[9]. https://www.analyticsvidhya.com/blog/2014/05/introduction-mapreduce/

[10]. https://www.cloudera.com/products/open-source/apache-hadoop.html