UNIT-I

Introduction to BIG Data:

Big Data is the term for collection of data sets so large and complex that it becomes difficult to process using on-hand database system tools or traditional data processing applications.

- Cost effective storage system for huge data sets.
- Provides ways to analyze information quickly and make decisions.
- Evaluation of customer needs and satisfaction.

Facts about Global Data:

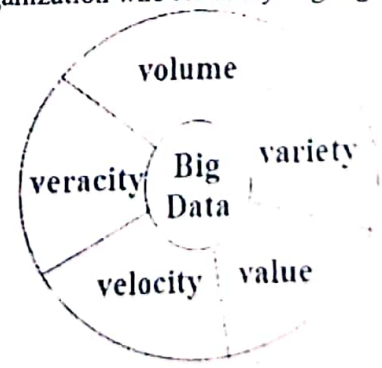
1. In 5 years there will be over 50 Billion smart connected devices in the word.
2. We create 2.5 Quintillion bytes of data every day.
3. 6.1 Billion global smart phone users by 2020.

Problems with Big Data:

- Storing exponentially growing huge datasets:
  1. Data generated in past two years is more than the previous history in total.
  2. The total digital data will grow to 44 Zettabytes approximately by 2020.
  3. About 1.7 MB of new information will be created every second for every person by 2020.
- Processing data have complex structures:
  1. Structured data where data is represented in organized format, example is RDBMS.
  2. Unstructured data where data is represented in un-organized format, example is multi-media files.
  3. Semi-structured data where data is represented in partially organized format, example XML.
- Cost to store Big Data is $25000 per TB per year.
- You need to copy all the data into the cluster and after processing write back to storage cluster.

5 Vs of Big Data:

1. **Volume:** Amount of data increasing day by day at very fast space. This data is generated by social media, humans, machines..etc.
2. **Variety:** There are many sources for big data. These sources may contain different categories of data such as structured data, unstructured data and semi structured data. 90% of the data generated is unstructured.
3. **Velocity:** The pace at which different sources generate the data every day. This flow of data is massive and continuous.
4. **Veracity:** Data in doubt or uncertainty of data due to inconsistency and incompleteness.
5. **Value:** Adding benefits to the organization who are analyzing big data.



Big Data Analytics Domains:

Big Data analysis is used in many Domains:

1. Web & E-Tailing
   E-Commerce companies such as Flipkart, Amazon, .. etc are all using big data analytics for their products.
2. Tele-communication
3. Government
4. Healthcare
5. Finance & Banking
6. Retail

Introduction to Hadoop:

Hadoop is a framework that allows us to store and process large data sets in parallel and distributed fashion

- ✓ For storage Hadoop uses HDFS (Hadoop Distributed File System).
- ✓ We can dump any kind of data across clusters. It will dump data and store it in distributed manner in clusters.
- ✓ Hadoop doesn't require high end servers.
- ✓ Hadoop uses commodity hardware (simple laptops and desktops of low cost which we used in our regular life).
- ✓ For processing Hadoop has YARN (Yet Another Resource Negotiator), it allows parallel processing of data in HDFS.

How Hadoop Resolves the Big Data Problem:

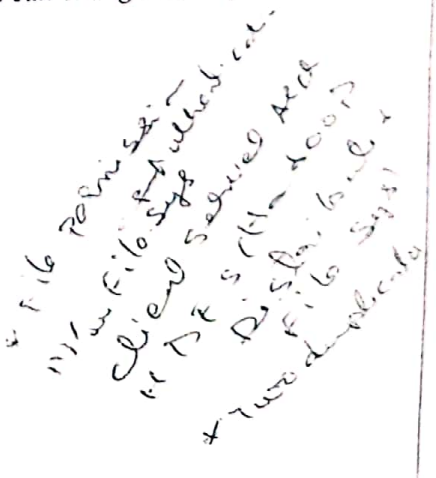
- ✣ With the introduction of Hadoop, we have HDFS which is the storage component of Hadoop.
- ✣ Hadoop helps you to store files in distributed manner and it can helps you in cut down the cost to $1000-$2000 per TB per Year.
- ✣ Hadoop has YARN (Yet another Resource Negotiator). With the help of YARN the data will be processed in parallel.
- ✣ Hadoop doesn't require to copy all the data into a computing cluster.
- ✣ If you want to increase the processing capability all you need to do is just increase the machines in the cluster.

HDFS (Hadoop Distributed File System):

- ⊥ The currently used file system is Network File System which was used in client server architecture.
- ⊥ Network File System uses high-end servers to store and process data. These high-end servers are cost expensive.
- ⊥ Hadoop File System was developed using distributed file system design in which the data is stored in commodity hardware.
- ⊥ Commodity hardware represents the systems we used in our daily routine.
- ⊥ The usage of commodity hardware makes HDFS cost effective.
- ⊥ HDFS stores very large amount of data in a easier way.
- ⊥ The data will be divided into number of blocks where each block will have a fixed size.
- ⊥ The block size in Hadoop 1.x version is 64 MB and in version 2.x is 128 MB.
- ⊥ These blocks of data will be stored in multiple systems across various clusters.
- ⊥ HDFS creates two duplicate copies of each and every file it is stored, i.e., if you a store a file about 1TB hadoop requires 3TB of storage.
- ⊥ This is called replication. The default replication factor is 2 and we can change this replication factor by configuring hadoop.
- ⊥ This replication helps to data loss in case of a failure.
- ⊥ HDFS allows parallel processing of data.

Features of HDFS:

- ✓ HDFS is suitable for the distributed storage and processing.
- ✓ Hadoop provides a command interface to interact with HDFS.
- ✓ HDFS provides file permissions and authentication.



HDFS Architecture:

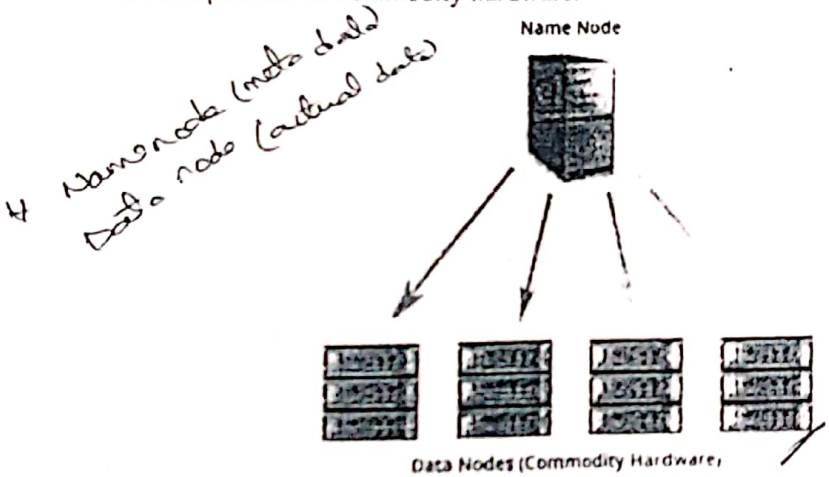
- HDFS architecture contains two major daemons. They are Name node and Data Node.

Name Node:

- Name Node is the Master daemon which is used to maintain and manage Data Nodes.
- Name Node records metadata, i.e., location of blocks stored, the size of the files, permission hierarchy of files etc.
- Name node receives heartbeat block report from all the Data Nodes.

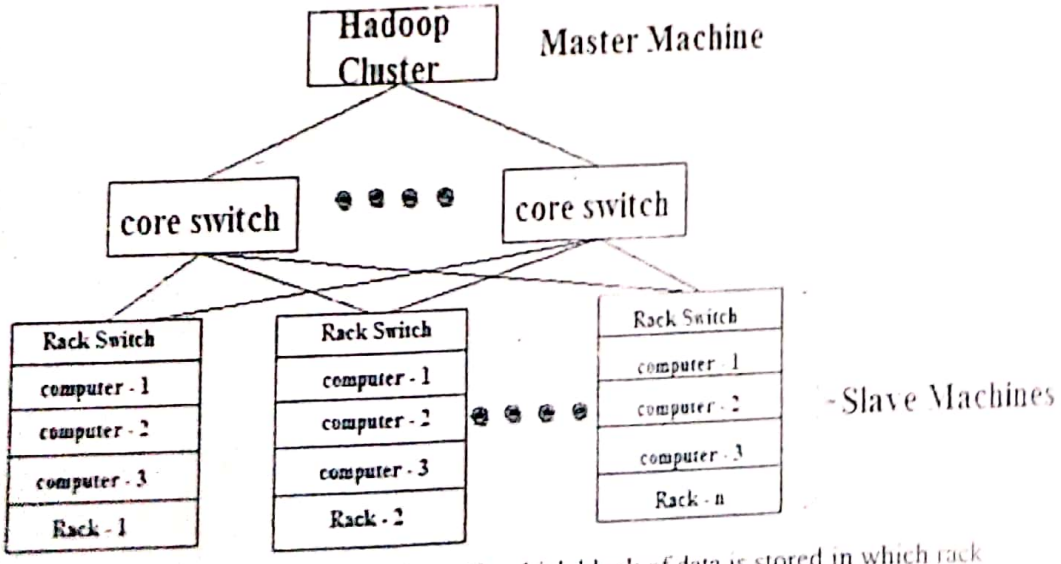
Data Node:

- Data Node is also known as slave daemon where actual data stored.
- Data Node serves read and write requests from clients.
- Data Nodes send a heartbeat to Name Node which is used to represent the node is alive.
- Data Nodes represent the commodity hardware.



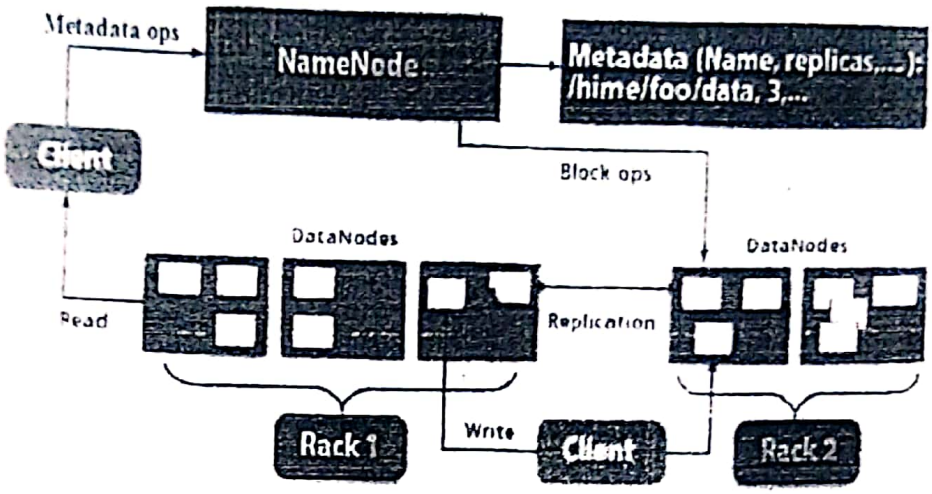
Hadoop Cluster Architecture:

- Hadoop cluster is represented as master slave topology.
- Hadoop cluster is also known as master machine which is connected to all slave nodes using core switch.
- The data nodes are stored in various racks.
- A Rack is a group of machines that are present physically at one particular location and are connected to each other.
- The racks are stored in different locations.



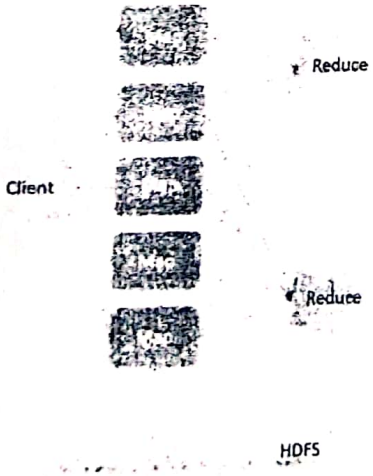
- Rack awareness algorithm is used to specify which block of data is stored in which rack.

HDFS Architecture



MAPREDUCING:

- Hadoop MapReduce is a softwae frameworkfor distributed processing of large data sets on computing clusters.
- MapReducing is a sub-project of the Apache Hadoop Project.
- Apache Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers.
- MapReduce is the core component for data processing in Hadoop framework.
- MapReduce helps to split the input data into a number of parts and run a program on all data parts parallel at a time.
- The term MapReduce refers to two separate and distinct tasks.
- The first is the map operation which takes a set of data and converts it into another set of data, where individual elements are divided into key-value pairs.
- The reduce operation combines those data tuples based on the key and accordingly modifies the value of the key.



Algorithm:

- Generally Mapreduce paradigm is based on sending the computer to where the data resides
- MapReducce program executes in three stages, namely map stage, shuffle stage and reduce stage

Map Stage:

- The map or mapper's job is to process the input data.
- The input data is in the form of file or directory and is stored in the Hadoop distributed file system.
- The input file is passed to the mapper function line by line.
- The mapper processes the data and creates several small chunks of data.

Reduce Stage:

- Reduce stage is the combination of shuffle stage and the reduce stage.
- The Reducer's job is to process the data that comes from the mapper.
- After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the clusters.
- The framework manages all the details of data passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.
- The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- Input and Output types of a **MapReduce job** = (Input) <k1, v1> → map → <k2, v2> → reduce → <k3, v3>(Output).

	Input	Output
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)