Q1) Discuss the characteristics and applications of Big Data.

Ans:- * Characteristics of Big Data:

i, VOLUME:

→ The name "Big Data" itself is related to a size which is enormous.

→ Volume is a huge amount of data.

→ To determine value of data, size of data plays a very crucial role. If the volume of data is very large then it is actually considered as Big Data.

→ That means whether a particular data can actually be considered as a Big Data or not, is dependent upon volume of data.

→ Hence while dealing with Big Data, it is necessar to consider a characteristic "Volume".

ii, VELOCITY:

→ It refers to high speed of accumulation of data.

→ In Big Data, velocity flows from in from sources like machine, networks, social media etc.

→ There is a massive and continous flow of data. This determines the potential of data that how fast the data is generated and processed to meet the demands.

→ Sampling data can help in dealing with issue like 'velocity'.

iii, VARIETY:

→ It refers to nature of data that is structured, semi-structured and unstructured data.

→ It also refers to heterogenous sources

→ It is basically the arrival of data from new sources that are both inside and outside of an enterprise.

* Applications of Big Data:

Following are few applications of Big Data:

i, Tracking customer Spending Habit, Shopping Behaviour
ii, Recommendations
iii, Smart Traffic System
iv, Secure Air Traffic System
v, Auto Driving Car
vi, Virtual Personal Assistant Tool
vii, IOT
viii, Education Sector
ix, Energy sector
x, Media & Entertainment Sector.

Q2) Explain the differences between RDBMS and Big Data.

Ans:- RDBMS is an information management system, which is based on a data model.

On the other hand, Hadoop is an open-source software framework used for storing data pertaining to Big Data concepts and running applications on group of hardware.

Following are the Differences between the two.

RDBMS	Hadoop
• Traditional row-column based databases, used for data storage, manipulation and retrieval	• An open-source software used for storing data & running applications concurrently.
• Structured data is mostly processed.	• Both structured & unstructured data is processed.
• Suitable for OLTP environment	• Suited for Big Data.
• Less scalable than Hadoop	• Highly scalable.
• Data scheme of RDBMS is static type.	• Data scheme of Hadoop is dynamic type.

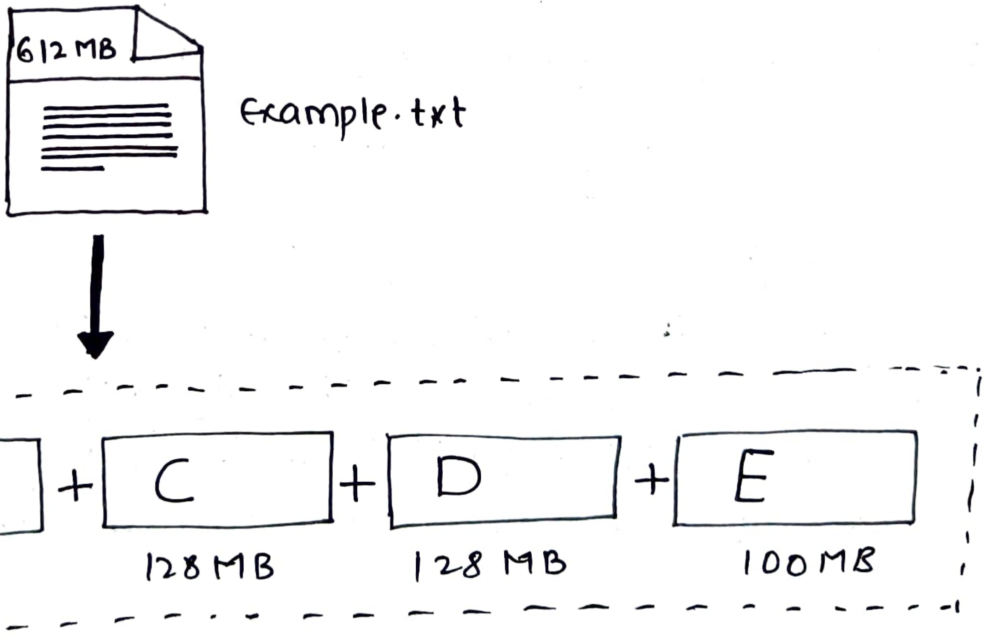
Q3) Explain blok size concept of HDFS with a neat diagra

Ans:- Hadoop is known for its reliable storage. Hadoop HDFS can store data of any size and format. HDFS (Hadoop Distributed File System) divides file into small size blocks called data blocks. These data blocks serve many advantages to Hadoop HDFS.

* Data Block in HDFS:

- Files in HDFS are broken into block-sized chunks called data blocks. These blocks are stored as independent units.
- The size of these HDFS data blocks is 128 mb by default. we can configure the block size as per our requirement by changing the "dbs. block. size" property in hdfs-site. xml.
- Hadoop distributes these blocks on different slave machines, and the master machine stores the metadata about blocks location.
- All the blocks of a file are of same except last one.

Following is an example.



Suppose we have a file size 612 MB, and we are using the default block configuration (128 mb). Theferore 5 blocks are created, the first four blocks are 128 mb in size, and fifth block is 100mb (128 * 4 + 100 = 612)

From the above example, we can conclude that :

i, A file in HDFS, smaller than ~~in~~ a single block does not occupy a full block size space of underlying storage

ii, Each file stored in HDFS doesn't need to be an exact multiple of configured block size.

* Advantages of Hadoop Data Blocks :

→ No limitations on the file size

→ Simplicity of storage subsystem

→ Fit well with replication for providing Fault Toleranc and High Availability.

→ Eliminating metadata concerns.

* Conclusion :

→ The files smaller than block size of a HDFS data block do not occupy the full block size.

→ Size of HDFS data block is large inorder to reduce the cost of seek and network traffic.

Q4) Write about History of Hadoop and explain how to load data into HDFS.

Ans:- * History of Hadoop:

→ The seeds of Hadoop were actually first planted in 2002 with the intention of building a better open source search engine.

→ It was started by Doug Cutting and Mike Cabarella. They called their initiative NUTCH, which inturn was a subproject of Apache. Doug Cutting founded all three projects : Lucene, Nutch, Hadoop

→ As we will see later, Apache Hadoop's goal was to be able to scale the entire web. It was around this time of October 2003, that Google published a paper describing the Google File System. Subsequently, in 2004 Google released another white paper on their MapReduce framework. Doug Cutting immediately saw the applicability of these technologies to NUTCH, ~~to~~ and thus implemented new framework based on that of Google's and ported Nutch to it.

→ Doug realised a dedicated project to flush out the new technologies was required to get to web scale. That is when Hadoop was born.

→ Yahoo hired Doug in Jan 2006 to work with a dedicated team in improving Hadoop ~~to~~ an open source project. Two years later, Hadoop achieved the status of an Apache Top Level project.

* Loading data into HDFS:

Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system. Follow the steps given below to insert/load the required file in the Hadoop File System.

STEP 01: You have to create an input directory

```
$ $HADOOP_HOME/bin/Hadoop fs -mkdir /user/input
```

STEP 02: Transfer and store a data file from local Systems to the Hadoop File system using put comman.

```
$ $HADOOP-HOME/bin/hadop fs -put /home/file.txt /user/input
```

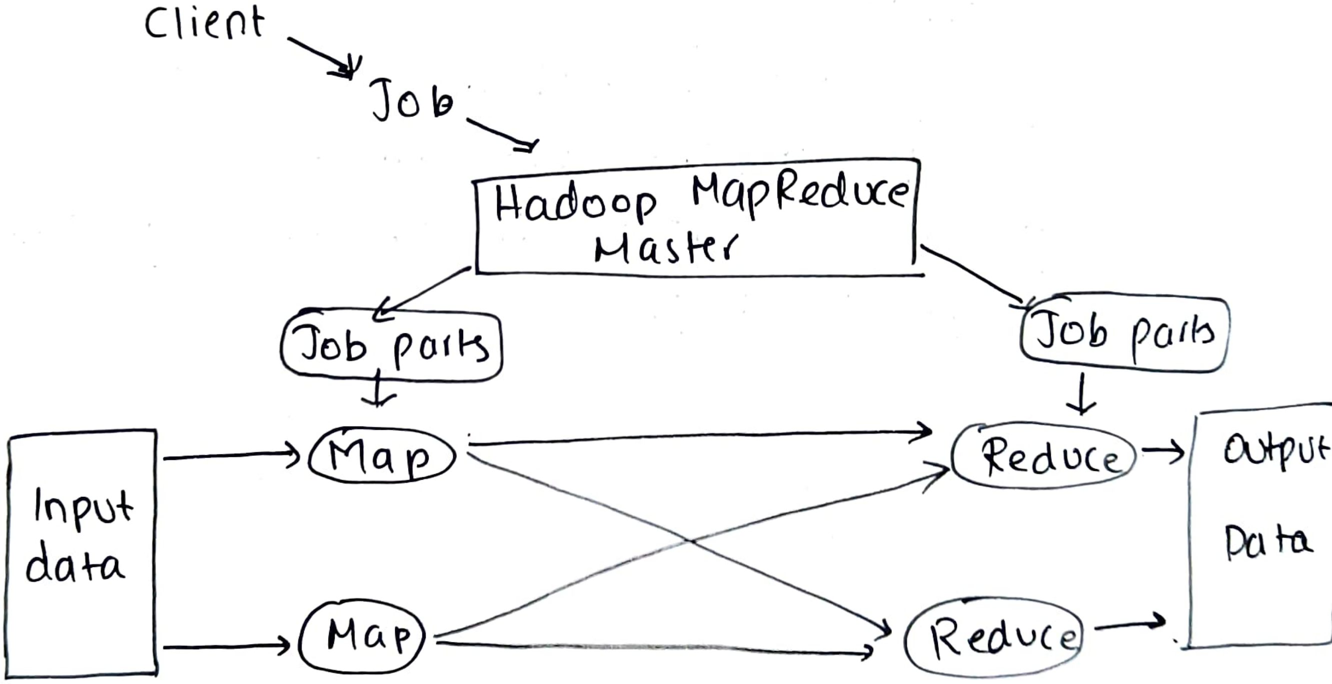
STEP 03: You can verify the file using "ls" command

```
$ $HADOOP_HOME/bin/hadoop fs -ls /user/input
```

Q5) Explain MapReduce Architecture.

Ans: MapReduce and HDFS are two major components of Hadoop which makes it so powerful and efficient to use. The purpose of MapReduce in Hadoop is to Map each of the jobs and then it will reduce it to equivalent tasks for providing less overhead over cluster network.

* MapReduce Architecture:



* Components of MapReduce Architecture:

i, Client: It is the one who brings the Job to MapReduce for processing. There can be multiple clients available that continously send jobs for processing to the Hadoop MapReduce Manager

ii, Job: It is the actual work that the client wanted to do which is comprised of so many smaller tasks that client wants to process or execute.

iii, Hadoop MapReduce Master: It divides particular job into subsequent job-parts

iv, Job Parts: The task or sub-jobs that are obtained after dividing the main job. the result of all the job-parts combined to produce final output.

v, Input Data: The data set that is fed to the MapReduce for processing.

vi, Output Data: The final result is obtained after the preprocessing.