INTRODUCTION TO HADOOP

Hadoop

- □ Hadoop is an open source framework by Apache Software Foundation which is written in Java for **storing and processing of huge datasets with the cluster of commodity hardware.**
- □ There are mainly two problems with the big data.
- □ First one is to store such a huge amount of data and the second one is to process that stored data.
- □ The traditional approach like RDBMS is not sufficient due to the heterogeneity of the data.
- □ So Hadoop comes as the solution to the problem of big data i.e. storing and processing the big data with some extra capabilities.

Features of Hadoop

□ **Economically Feasible:** It is cheaper to store data and process it than it was in the traditional approach. Since the actual machines used to store data are only commodity hardware.

□ **Easy to Use:** The projects or set of tools provided by Apache Hadoop are easy to work upon in order to analyze complex data sets.

□ **Open Source:** Since Hadoop is distributed as an open source software under Apache License, so one does not need to pay for it, just download it and use it.

□ **Fault Tolerance:** Since Hadoop stores three copies of data, so even if one copy is lost because of any commodity hardware failure, the data is safe. Moreover, as Hadoop version 3 has multiple name nodes, so even the single point of failure of Hadoop has also been removed.

□ **Scalability:** Hadoop is highly scalable in nature. If one needs to scale up or scale down the cluster, one only needs to change the number of commodity hardware in the cluster.

□ **Distributed Processing:** HDFS and Map Reduce ensures distributed storage and processing of the data.

□ **Locality of Data:** This is one of the most alluring and promising features of Hadoop. In Hadoop, to process a query over a data set, instead of bringing the data to the local computer we send the query to the server and fetch the final result from there. This is called data locality.

Advantages and Disadvantages of Hadoop

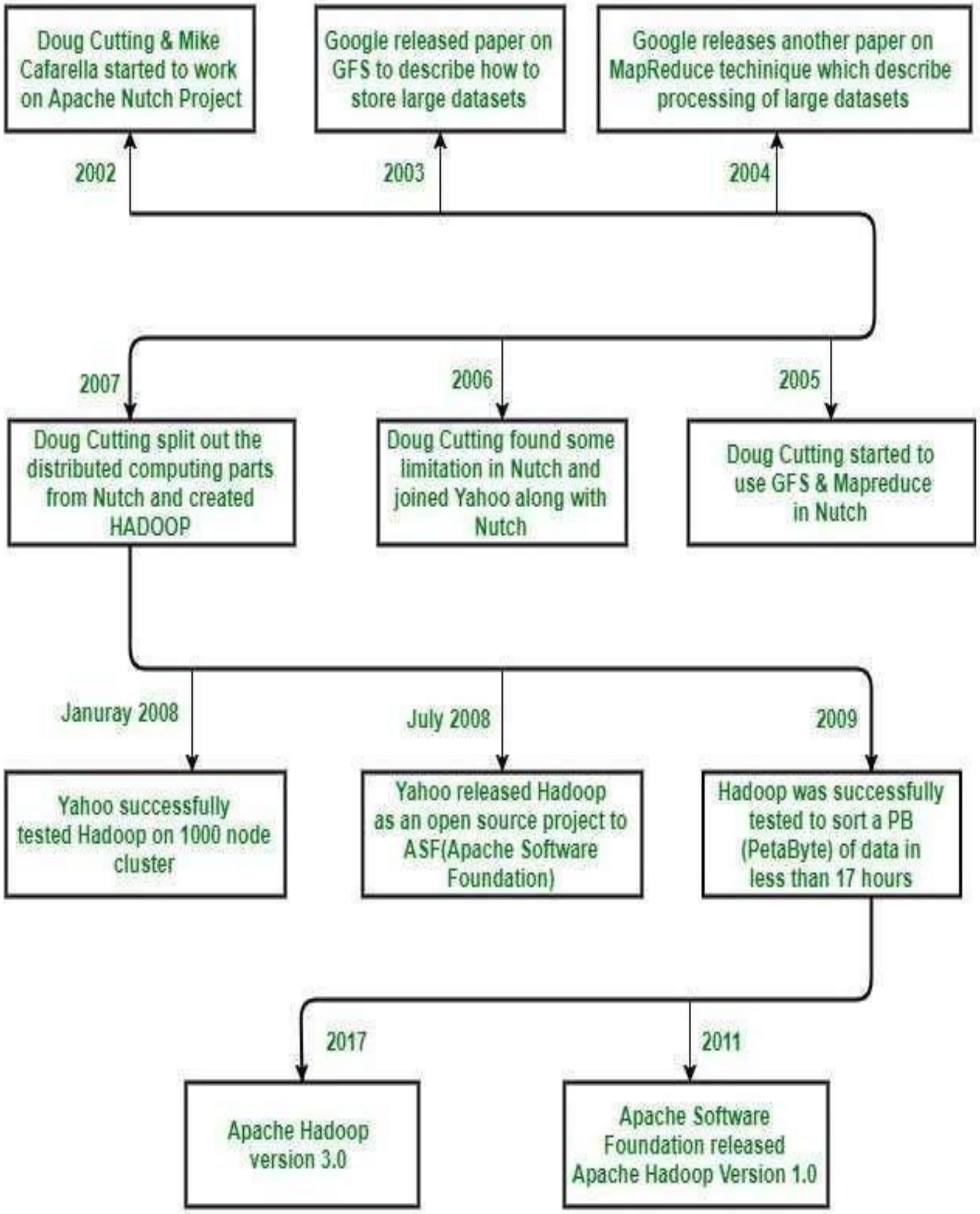
Advantages:

- □ Ability to store a large amount of data.
- □ High flexibility.
- □ Cost effective.
- □ High computational power.
- □ Tasks are independent.
- □ Linear scaling.

Disadvantages:

- □ Not very effective for small data.
- □ HDFS can not mounted directly by an existing OS.
- □ Has stability issues.
- □ Security concerns.
- □ Vulnerable by nature.
- □ No Real time processing

HADOOP History



2002

Doug Cutting & Mike Cafarella started to work on Apache Nutch Project

2003

Google released paper on GFS to describe how to store large datasets

2004

Google releases another paper on MapReduce techinique which describe processing of large datasets

2005

Doug Cutting started to use GFS & Mapreduce in Nutch

2006

Doug Cutting found some limitation in Nutch and joined Yahoo along with Nutch

2007

Doug Cutting split out the distributed computing parts from Nutch and created HADOOP

Januray 2008

Yahoo successfully tested Hadoop on 1000 node cluster

July 2008

Yahoo released Hadoop as an open source project to ASF(Apache Software Foundation)

2009

Hadoop was successfully tested to sort a PB (PetaByte) of data in less than 17 hours

2011

Apache Software Foundation released Apache Hadoop Version 1.0

2017

Apache Hadoop version 3.0

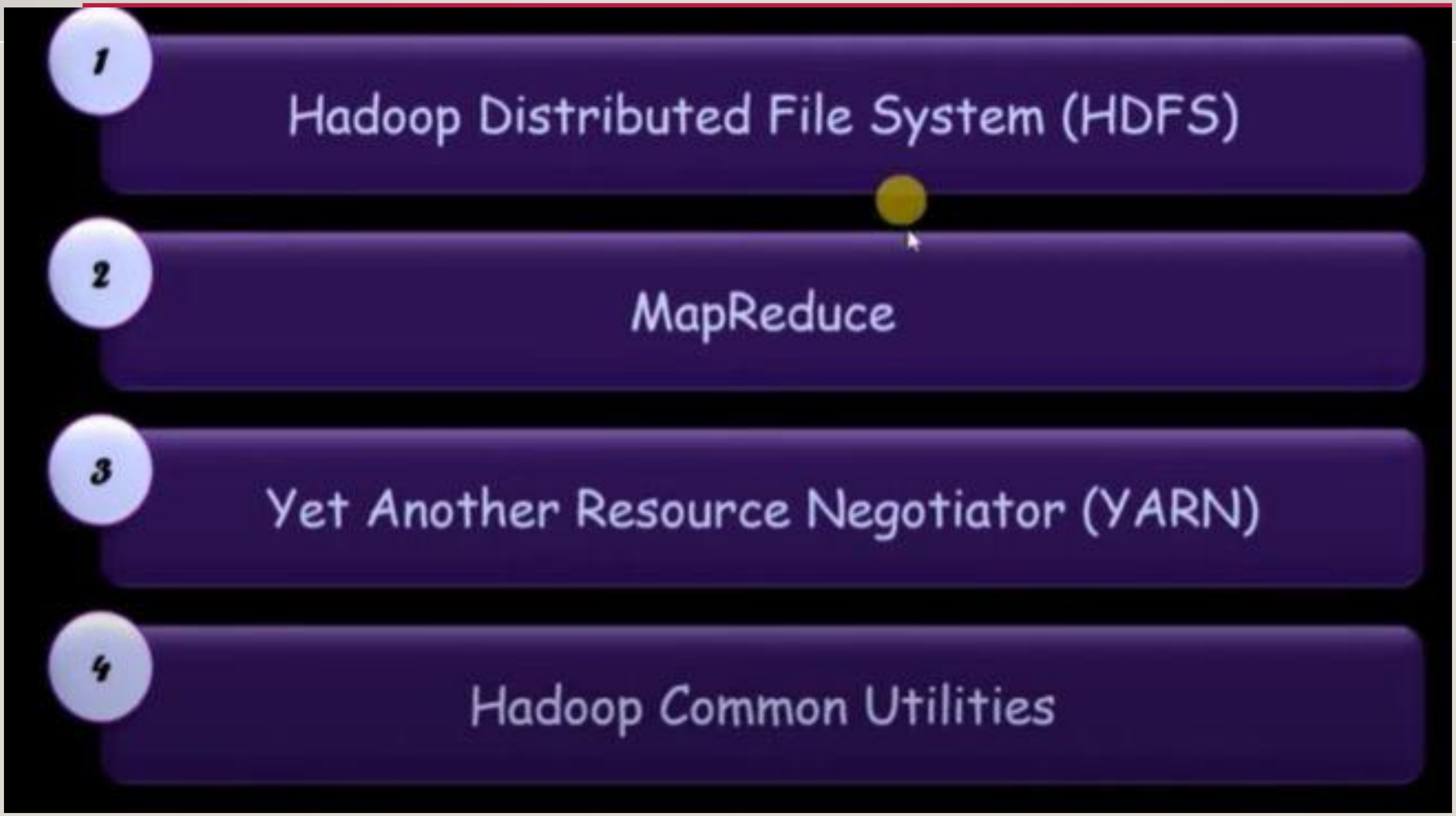
Hadoop Versions

Hadoop 1: This is the first and most basic version of Hadoop. It includes Hadoop Common, Hadoop Distributed File System (HDFS), and Map Reduce.

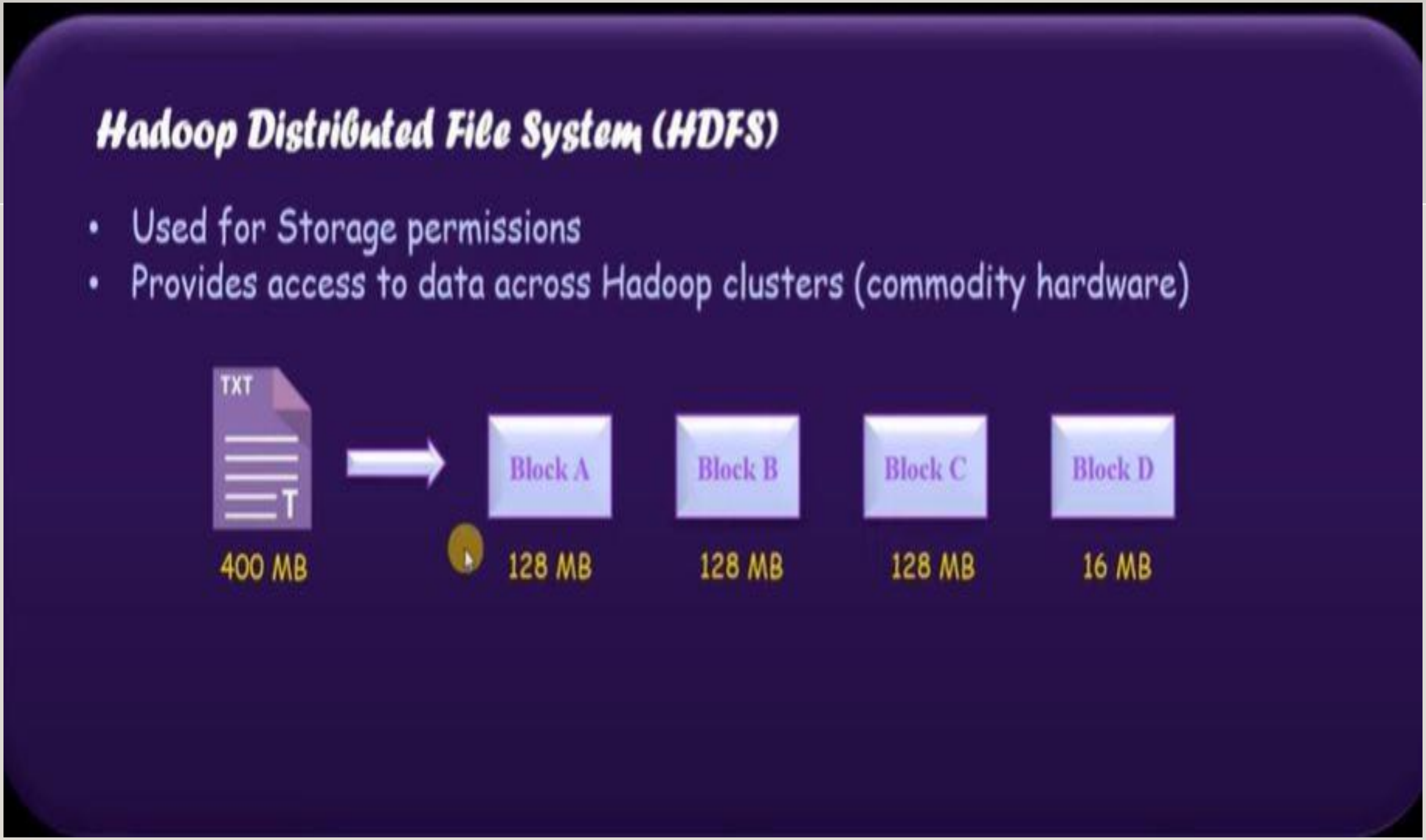
Hadoop 2: The only difference between Hadoop 1 and Hadoop 2 is that Hadoop 2 additionally contains YARN (Yet Another Resource Negotiator). YARN helps in resource management and task scheduling through its two daemons namely job tracking and progress monitoring.

Hadoop 3: This is the recent version of Hadoop. Along with the merits of the first two versions, Hadoop 3 has one most important merit. It has resolved the issue of single point failure by having multiple name nodes. Various other advantages like erasure coding, use of GPU hardware and Dockers makes it superior to the earlier versions of Hadoop.

CORE COMPONENTS OF HADOOP

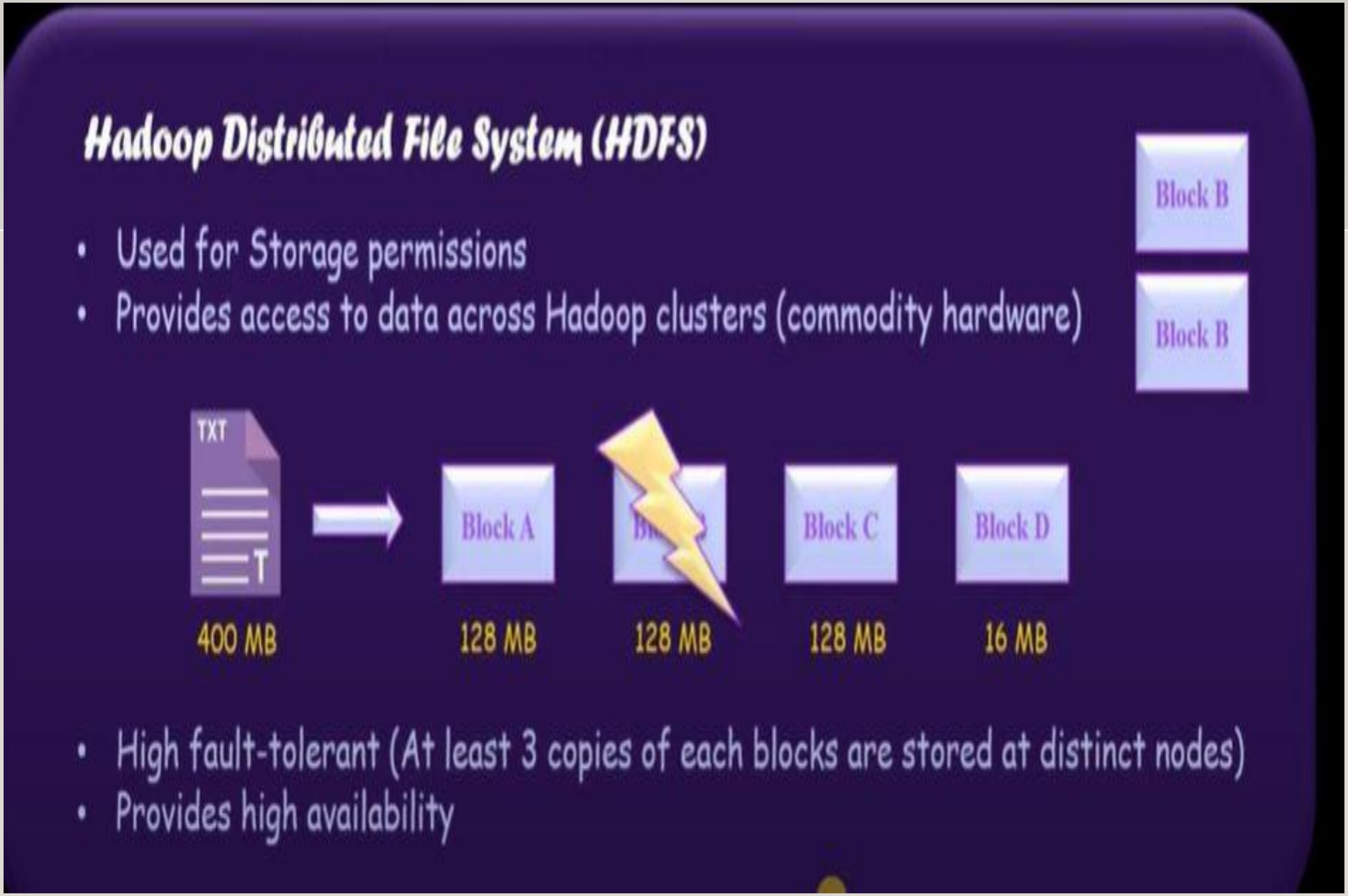


1. Hadoop Distributed File System (HDFS)
2. MapReduce
3. Yet Another Resource Negotiator (YARN)
4. Hadoop Common Utilities



Hadoop Distributed File System (HDFS)

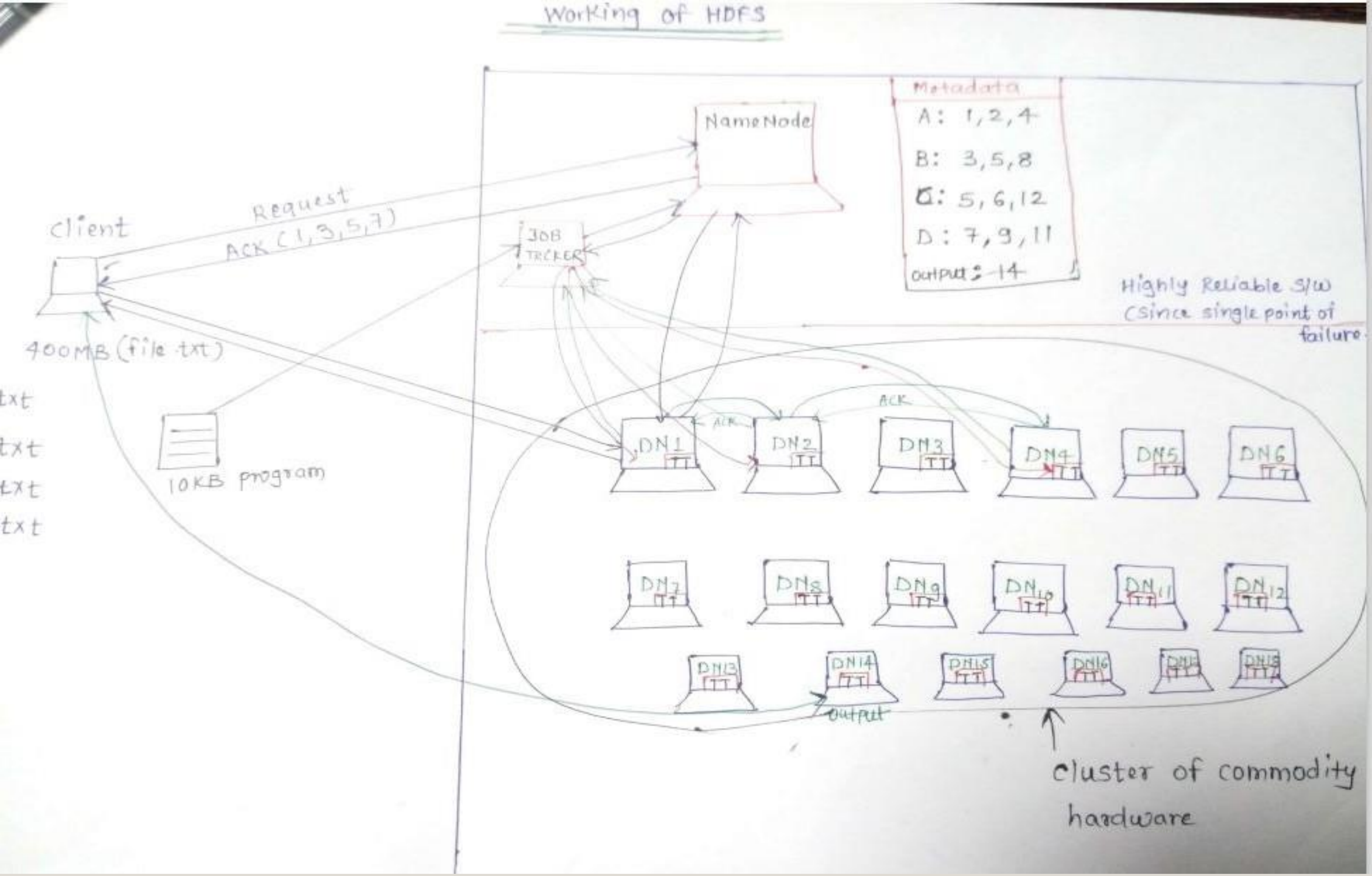
- Used for Storage permissions
- Provides access to data across Hadoop clusters (commodity hardware)



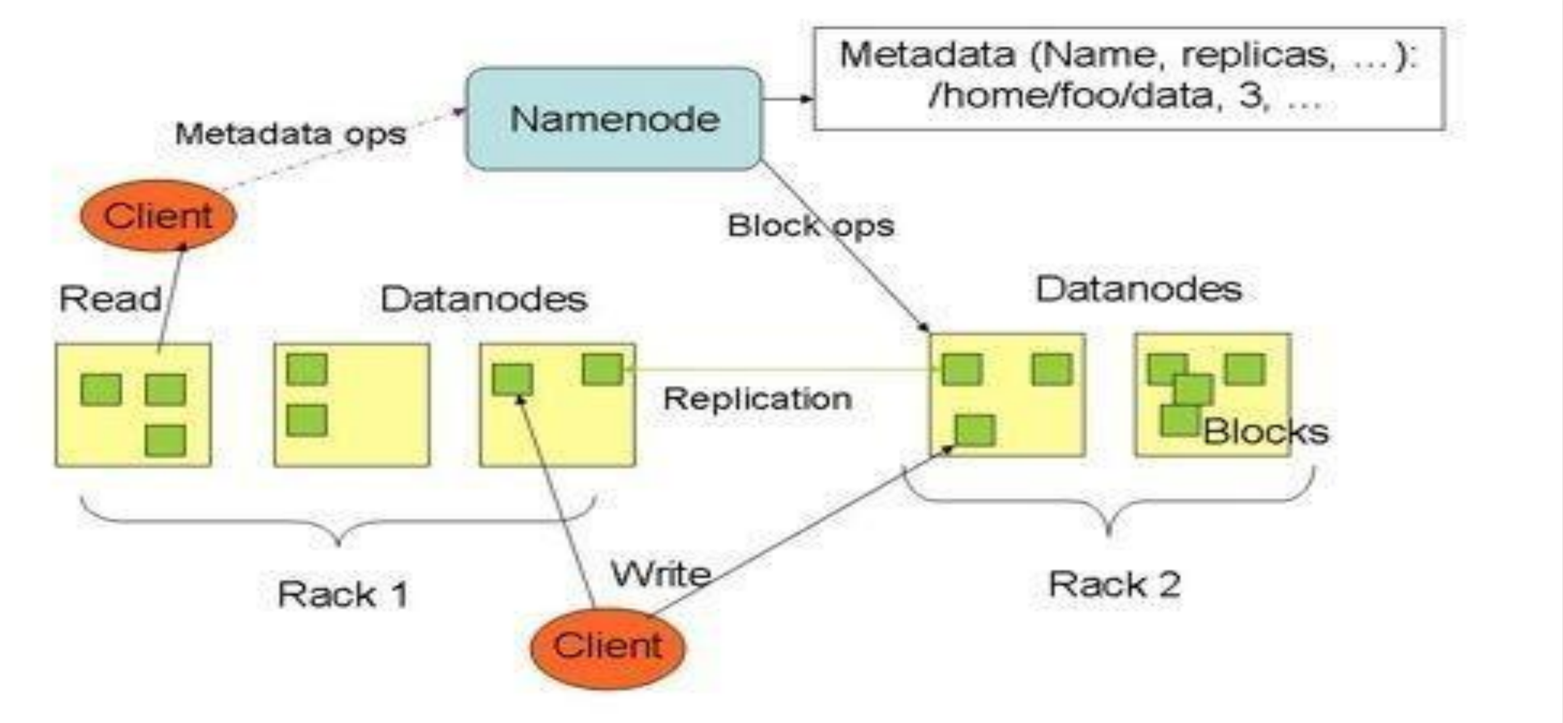
Hadoop Distributed File System (HDFS)

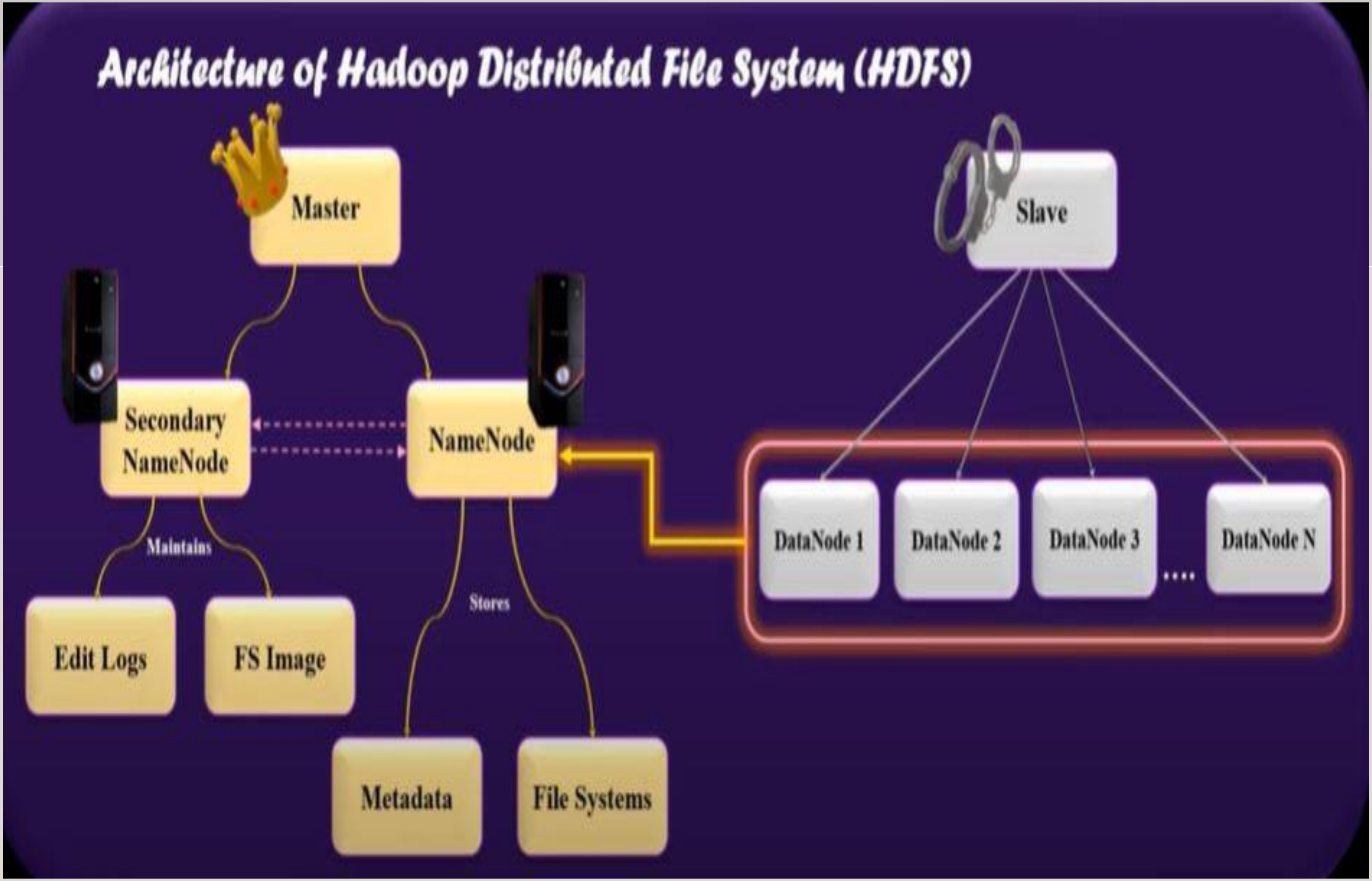
- Used for Storage permissions
- Provides access to data across Hadoop clusters (commodity hardware)

- High fault-tolerant (At least 3 copies of each blocks are stored at distinct nodes)
- Provides high availability

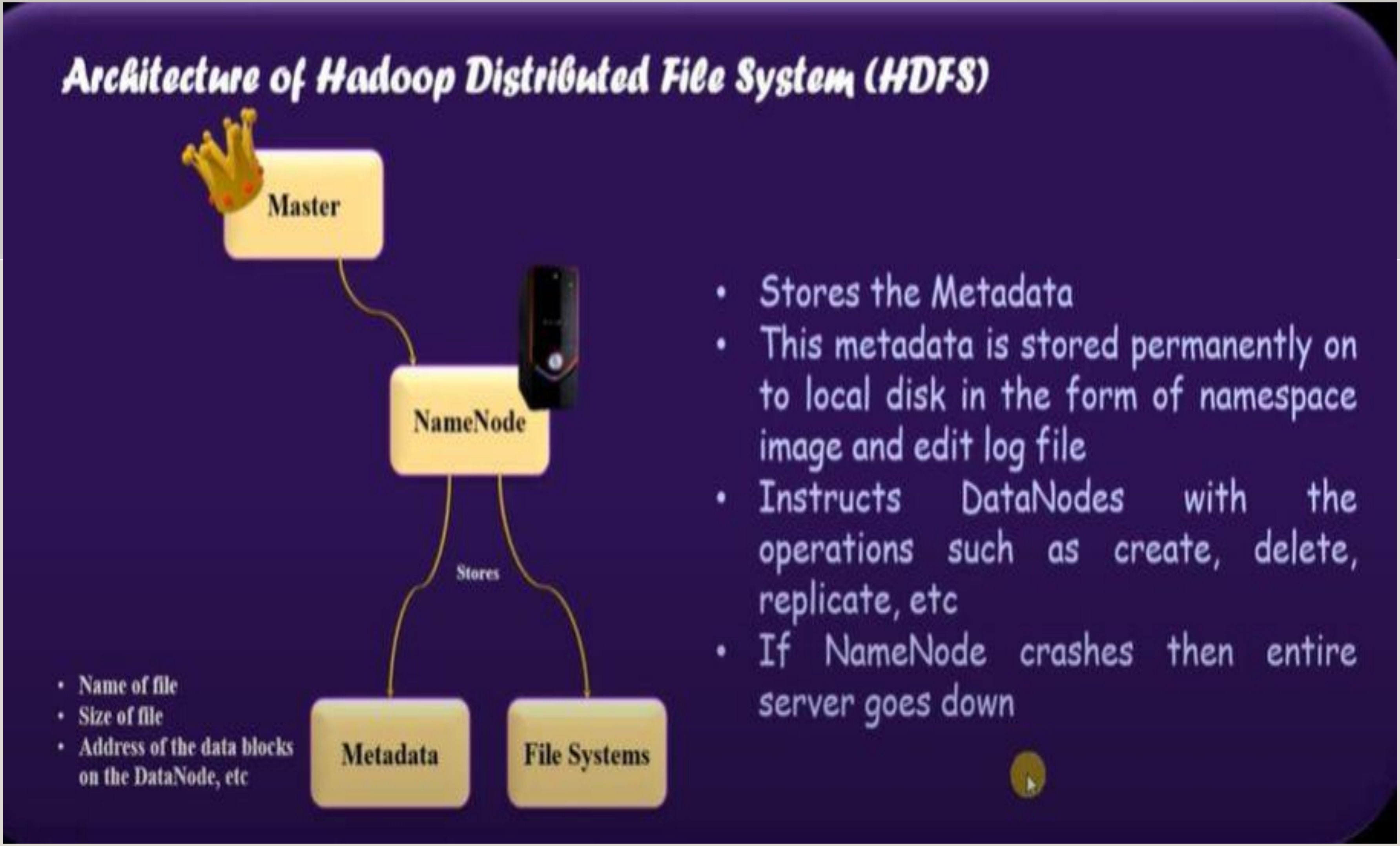

Working of HDFS
NameNode
Metadata
A: 1,2,4
B: 3,5,8
C: 5,6,12
D: 7,9,11
output: 14
Request
ACK (1,3,5,7)
Client
JOB TRCKER
400MB (file txt)
txt
txt
txt
txt
10KB program
Highly Reliable S/W (since single point of failure
ACK
ACK
DN1
DN2
DN3
DN4
DN5
DN6
DN7
DN8
DN9
DN10
DN11
DN12
DN13
DN14
DN15
DN16
DN17
DN18
TT
output
cluster of commodity hardware

REPLICATION AND READ-WRITES IN HDFS





Architecture of Hadoop Distributed File System (HDFS)



Architecture of Hadoop Distributed File System (HDFS)

Master

NameNode

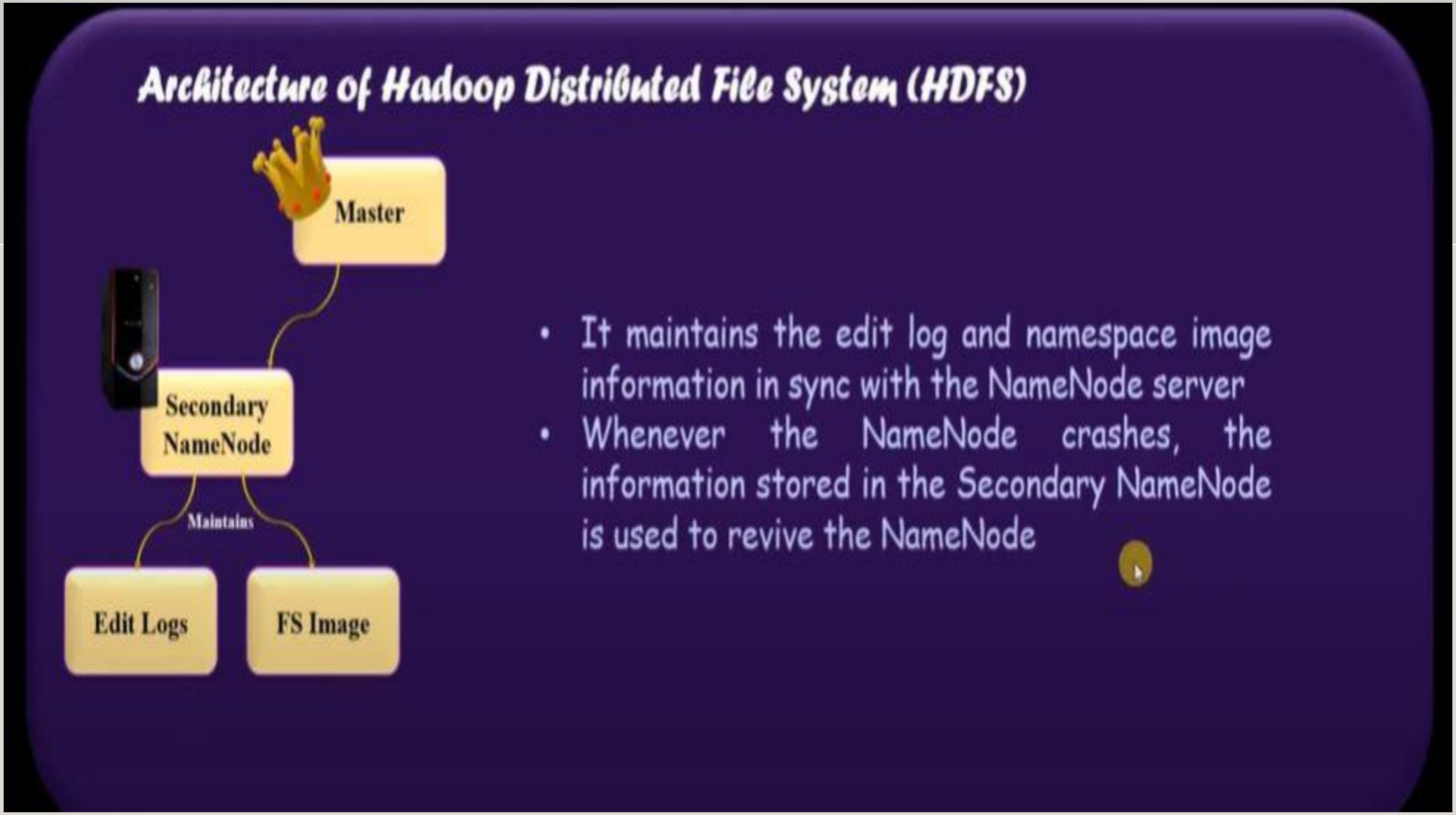
Stores

Metadata

File Systems

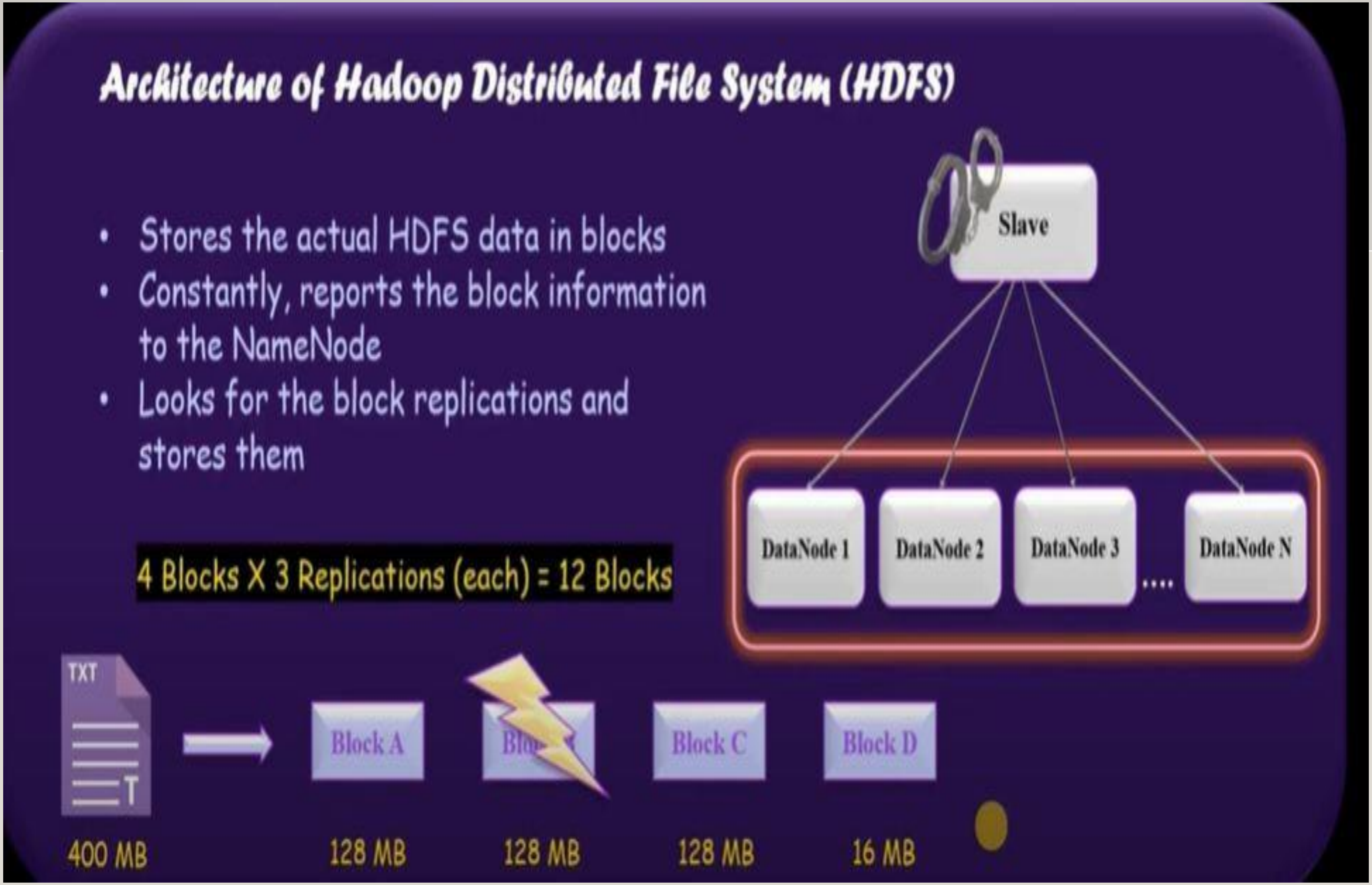
- Name of file
- Size of file
- Address of the data blocks on the DataNode, etc

- Stores the Metadata
- This metadata is stored permanently on to local disk in the form of namespace image and edit log file
- Instructs DataNodes with the operations such as create, delete, replicate, etc
- If NameNode crashes then entire server goes down



Architecture of Hadoop Distributed File System (HDFS)

- It maintains the edit log and namespace image information in sync with the NameNode server
- Whenever the NameNode crashes, the information stored in the Secondary NameNode is used to revive the NameNode



Architecture of Hadoop Distributed File System (HDFS)

- Stores the actual HDFS data in blocks
- Constantly, reports the block information to the NameNode
- Looks for the block replications and stores them

4 Blocks X 3 Replications (each) = 12 Blocks

Slave

DataNode 1

DataNode 2

DataNode 3

....

DataNode N

TXT

T

Block A

Blo

Block C

Block D

400 MB

128 MB

128 MB

128 MB

16 MB

THANK YOU!