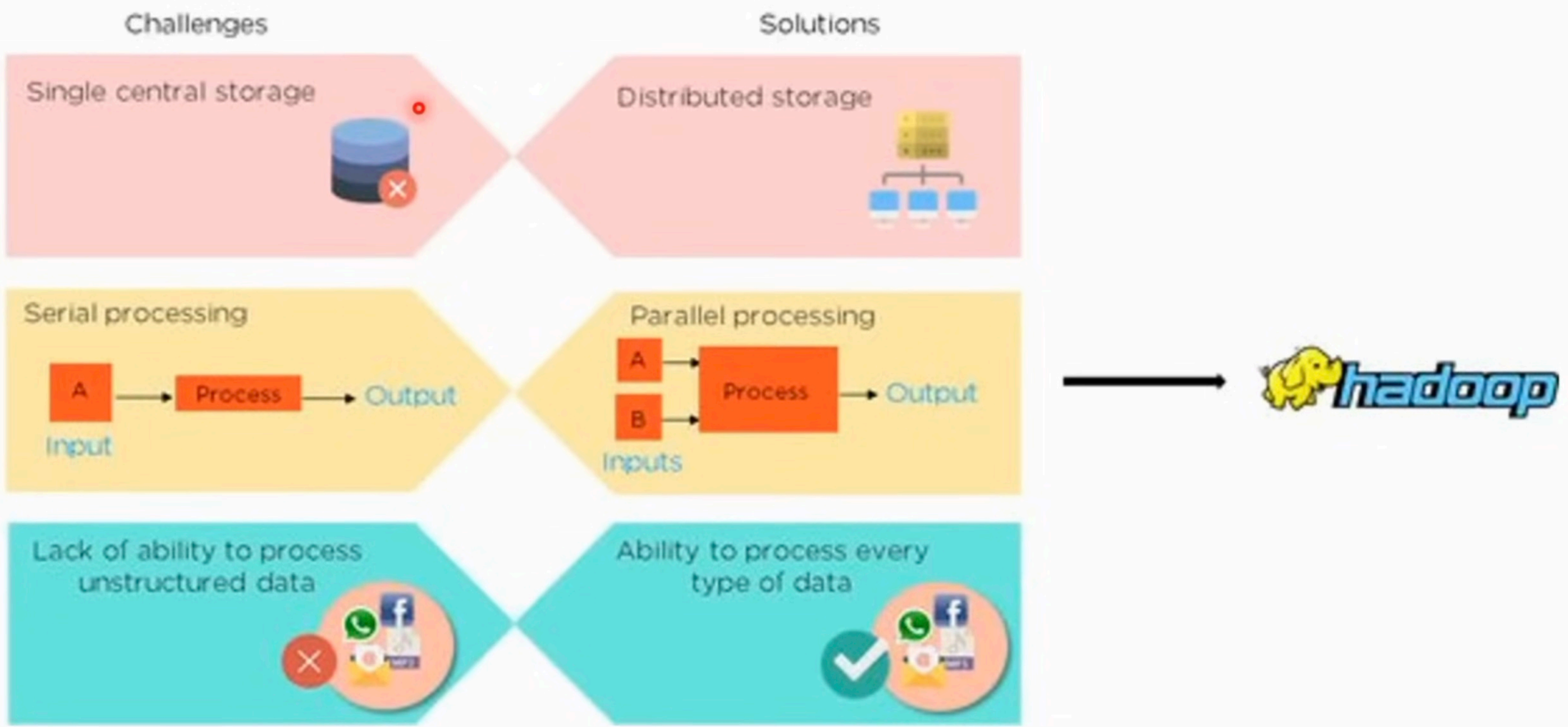
Hadoop is an open source, Java based framework used for storing and processing big data.

Hadoop uses the MapReduce programming model for faster storage and retrieval of data from its nodes.

The framework is managed by Apache Software Foundation and is licensed under the Apache License 2.0.

It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more.


Challenges
Solutions
Single central storage
Distributed storage
Serial processing
A
Process
Output
Input
Parallel processing
A
B
Process
Output
Inputs
Lack of ability to process unstructured data
Ability to process every type of data
hadoop

Modules of Hadoop

1.HDFS: Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.

2.Yarn: Yet another Resource Negotiator is used for job scheduling and manage the cluster.

3.Map Reduce: This is a framework which helps Java programs to do the parallel computation on data using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair. The output of Map task is consumed by reduce task and then the out of reducer gives the desired result.

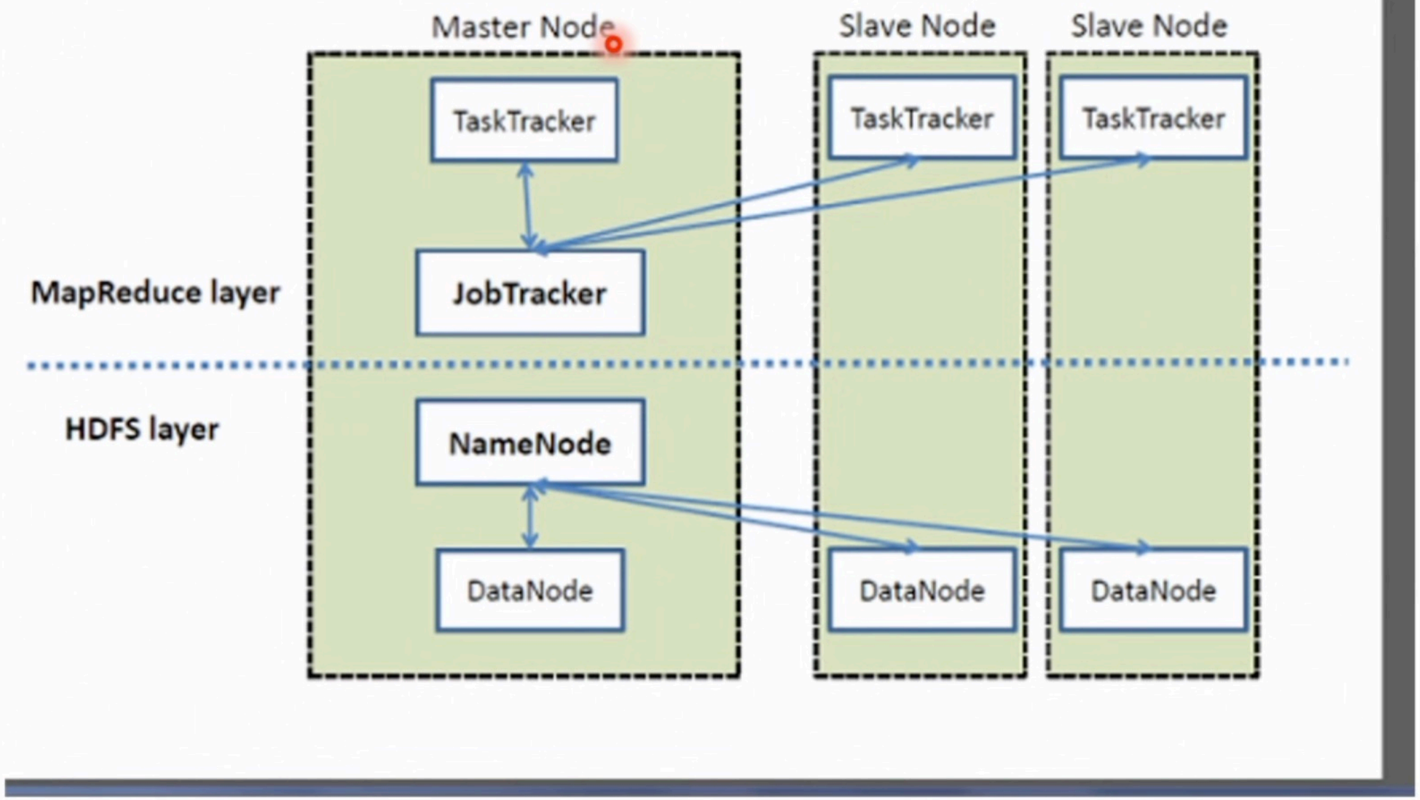
4.Hadoop Common: These Java libraries are used to start Hadoop and are used by other Hadoop modules.

Hadoop Architecture ?

The Hadoop Distributed File System (HDFS) is a distributed file system for Hadoop. It contains a master/slave architecture. This architecture consist of a single NameNode performs the role of master, and multiple DataNodes performs the role of a slave.

Both NameNode and DataNode are capable enough to run on commodity machines. The Java language is used to develop HDFS. So any machine that supports Java language can easily run the NameNode and DataNode software.

High Level Architecture of Hadoop



NameNode

•It is a single master server exist in the HDFS cluster.
•As it is a single node, it may become the reason of single point failure.
•It manages the file system namespace by executing an operation like the opening, renaming and closing the files.
•It simplifies the architecture of the system.

DataNode

•The HDFS cluster contains multiple DataNodes.
•Each DataNode contains multiple data blocks.
•These data blocks are used to store data.
•It is the responsibility of DataNode to read and write requests from the file system's clients.
•It performs block creation, deletion, and replication upon instruction from the NameNode.

Job Tracker

•The role of Job Tracker is to accept the MapReduce jobs from client and process the data by using NameNode.
•In response, NameNode provides metadata to Job Tracker.

Task Tracker

•It works as a slave node for Job Tracker.
•It receives task and code from Job Tracker and applies that code on the file. This process can also be called as a Mapper.

MapReduce Layer

The MapReduce comes into existence when the client application submits the MapReduce job to Job Tracker. In response, the Job Tracker sends the request to the appropriate Task Trackers. Sometimes, the TaskTracker fails or time out. In such a case, that part of the job is rescheduled.

TRUE ENGINEER

What is HDFS

Hadoop comes with a distributed file system called HDFS.

In HDFS data is distributed over several machines and replicated to ensure their durability to failure and high availability to parallel application.

It is cost effective as it uses commodity hardware. It involves the concept of blocks, data nodes and node name.

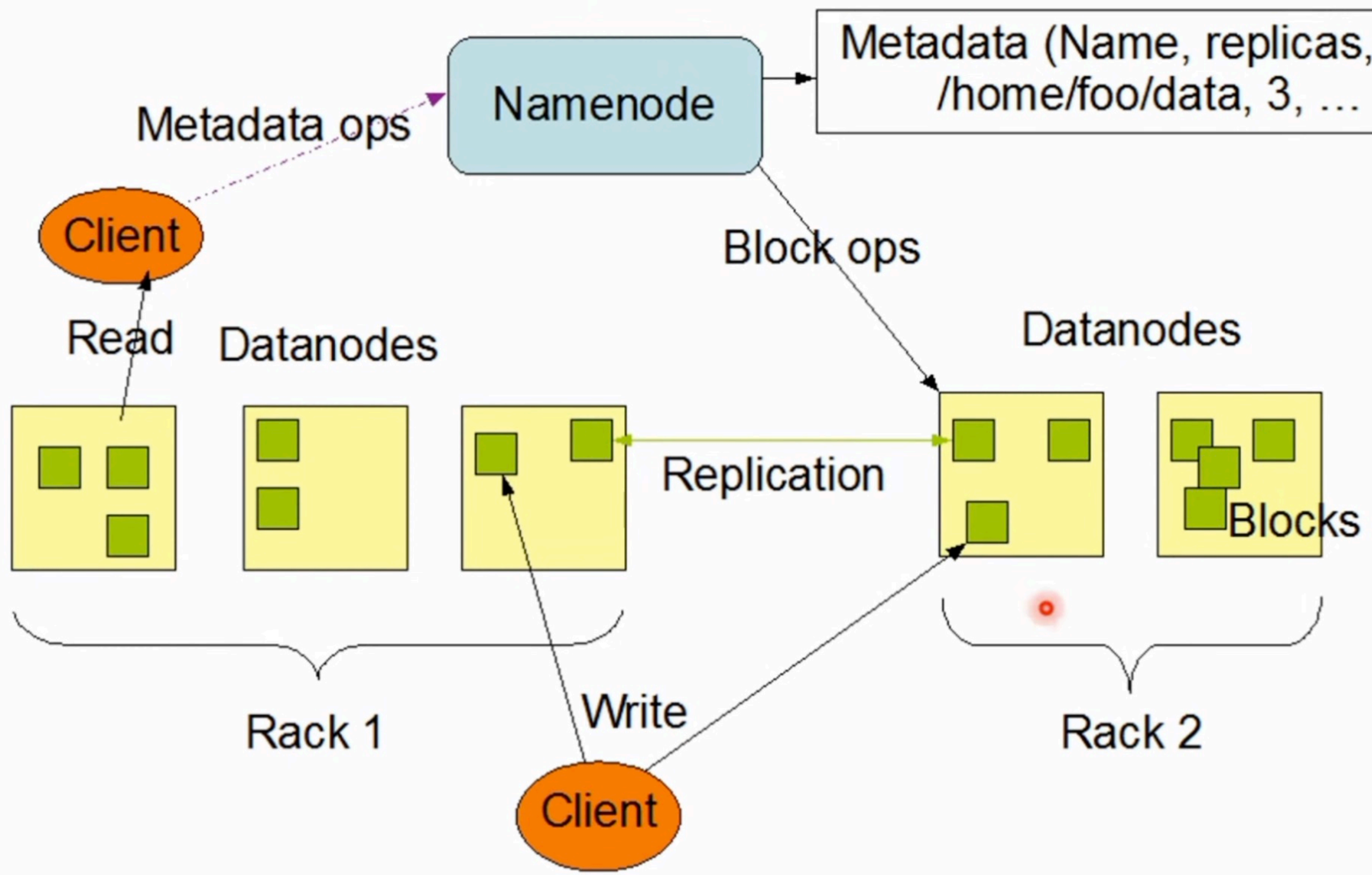
Where to use HDFS

- **Very Large Files:** Files should be of hundreds of megabytes, gigabytes or more.
- **Streaming Data Access:** The time to read whole data set is more important than latency in reading the first. HDFS is built on write-once and read-many-times pattern.
- **Commodity Hardware:** It works on low cost hardware.

Where not to use HDFS

- **Low Latency data access:** Applications that require very less time to access the first data should not use HDFS as it is giving importance to whole data rather than time to fetch the first record.
- **Lots Of Small Files:** The name node contains the metadata of files in memory and if the files are small in size it takes a lot of memory for name node's memory which is not feasible.
- **Multiple Writes:** It should not be used when we have to write multiple times.

HDFS Architecture



1.Name Node:

HDFS works in master-worker pattern where the name node acts as master.

Name Node is controller and manager of HDFS as it knows the status and the metadata of all the files in HDFS; the metadata information being file permission, names and location of each block.

The metadata are small, so it is stored in the memory of name node, allowing faster access to data.

Moreover the HDFS cluster is accessed by multiple clients concurrently, so all this information is handled by a single machine. The file system operations like opening, closing, renaming etc. are executed by it.

1.Blocks:

A Block is the minimum amount of data that it can read or write.

HDFS blocks are 128 MB by default and this is configurable.

Files in HDFS are broken into block-sized chunks, which are stored as independent units.

Unlike a file system, if the file is in HDFS is smaller than block size, then it does not occupy full blocks size,

i.e. 5 MB of file stored in HDFS of block size 128 MB takes 5MB of space only.

The HDFS block size is large just to minimize the cost of seek.

1.Data Node:

They store and retrieve blocks when they are told to; by client or name node.

They report back to name node periodically, with list of blocks that they are storing.

The data node being a commodity hardware also does the work of block creation, deletion and replication as stated by the name node.