Dr. Ambarish G. Mohapatra

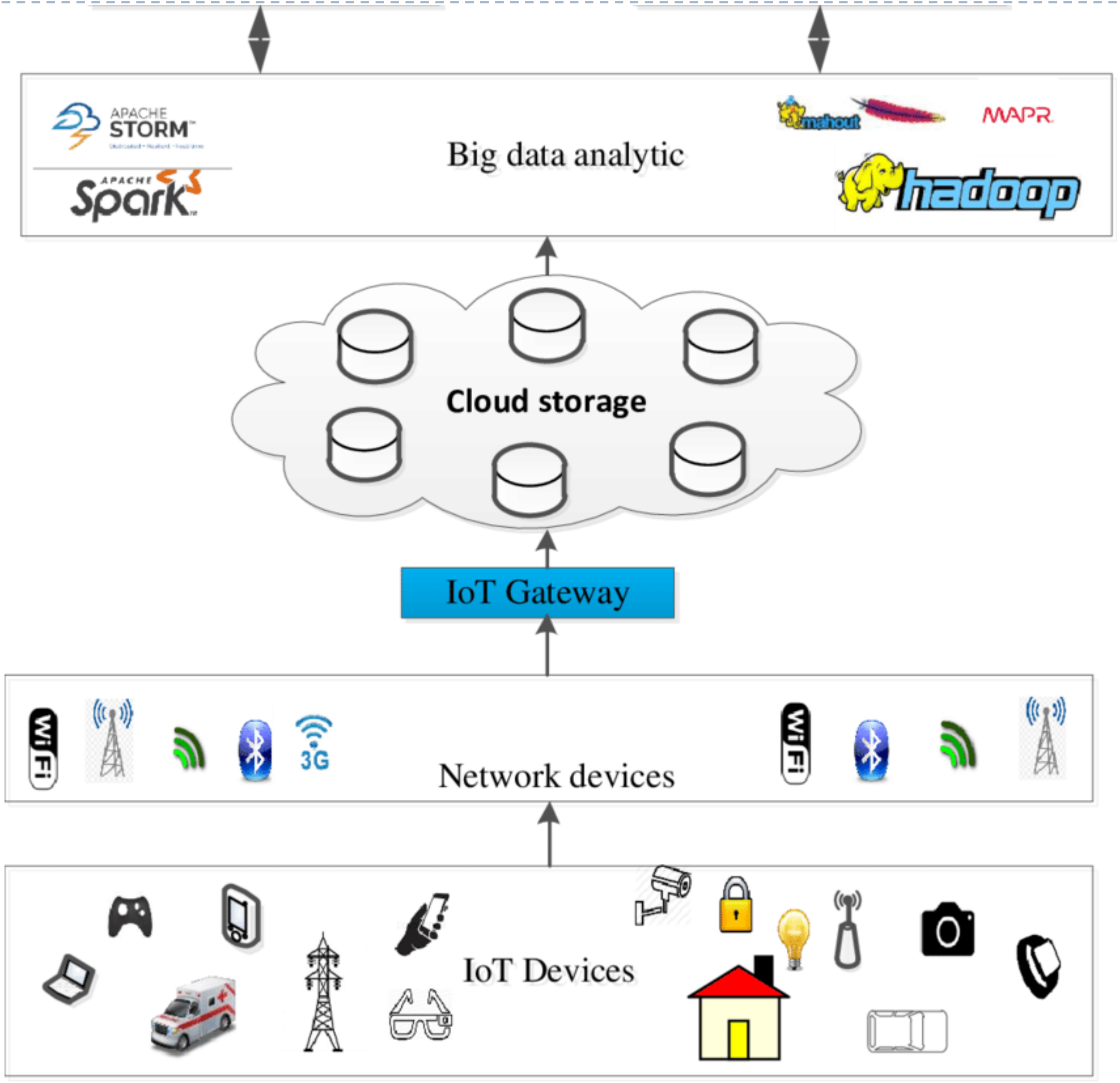
# IOT & Applications

## Module – 5
## Class-1

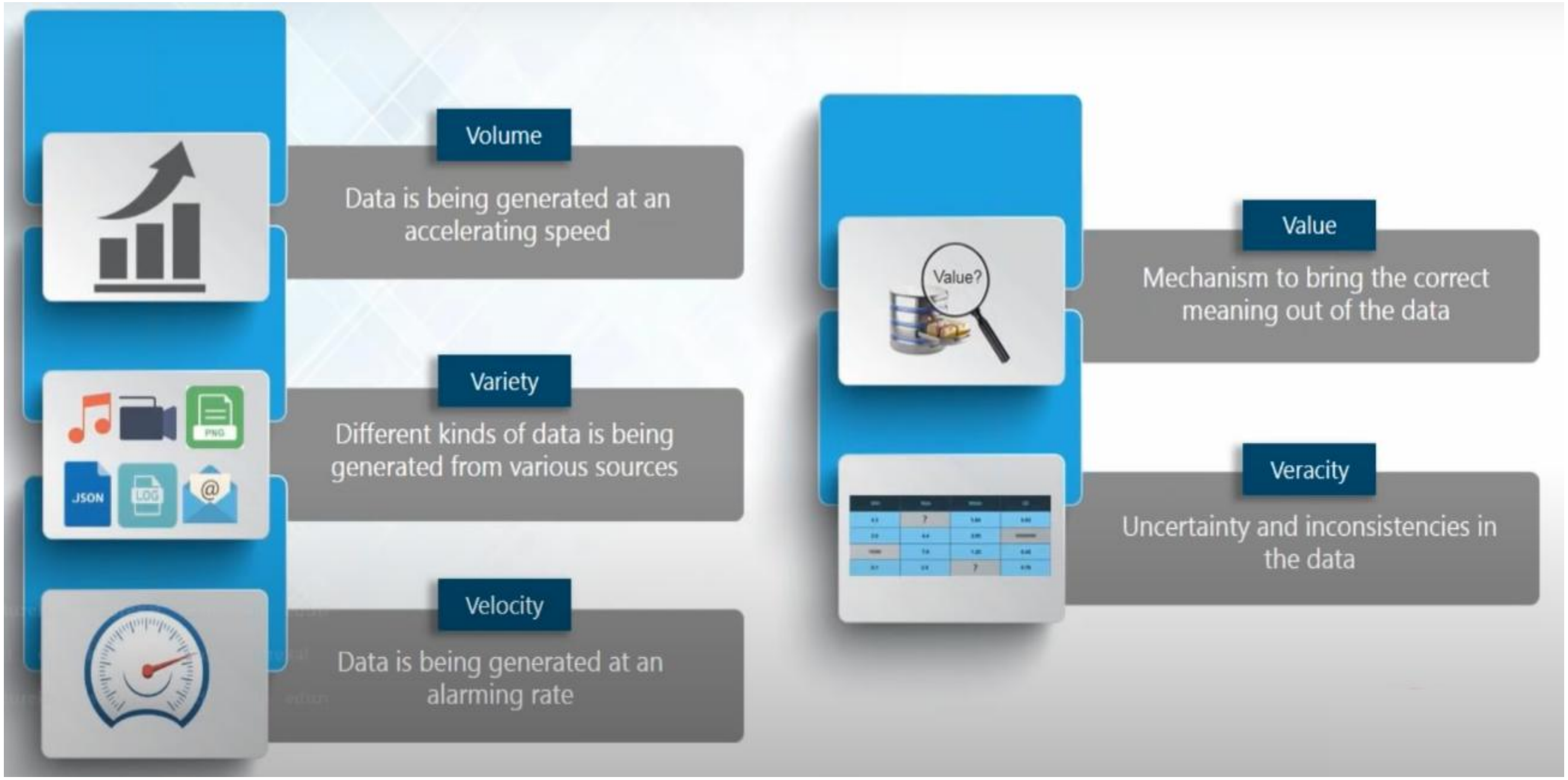
**Introduction to Data Analytics**: Introduction to Apache Hadoop: MapReduce, Hadoop ClusterSetup, YARN, Oozie, Spark, Kafka, Apache Storm

Silicon ...beyond teaching | Silicon Institute of Technology Bhubaneswar

## IoT and BigData



## 5'Vs of BigData



Volume

Data is being generated at an accelerating speed

Variety

Different kinds of data is being generated from various sources

Velocity

Data is being generated at an alarming rate

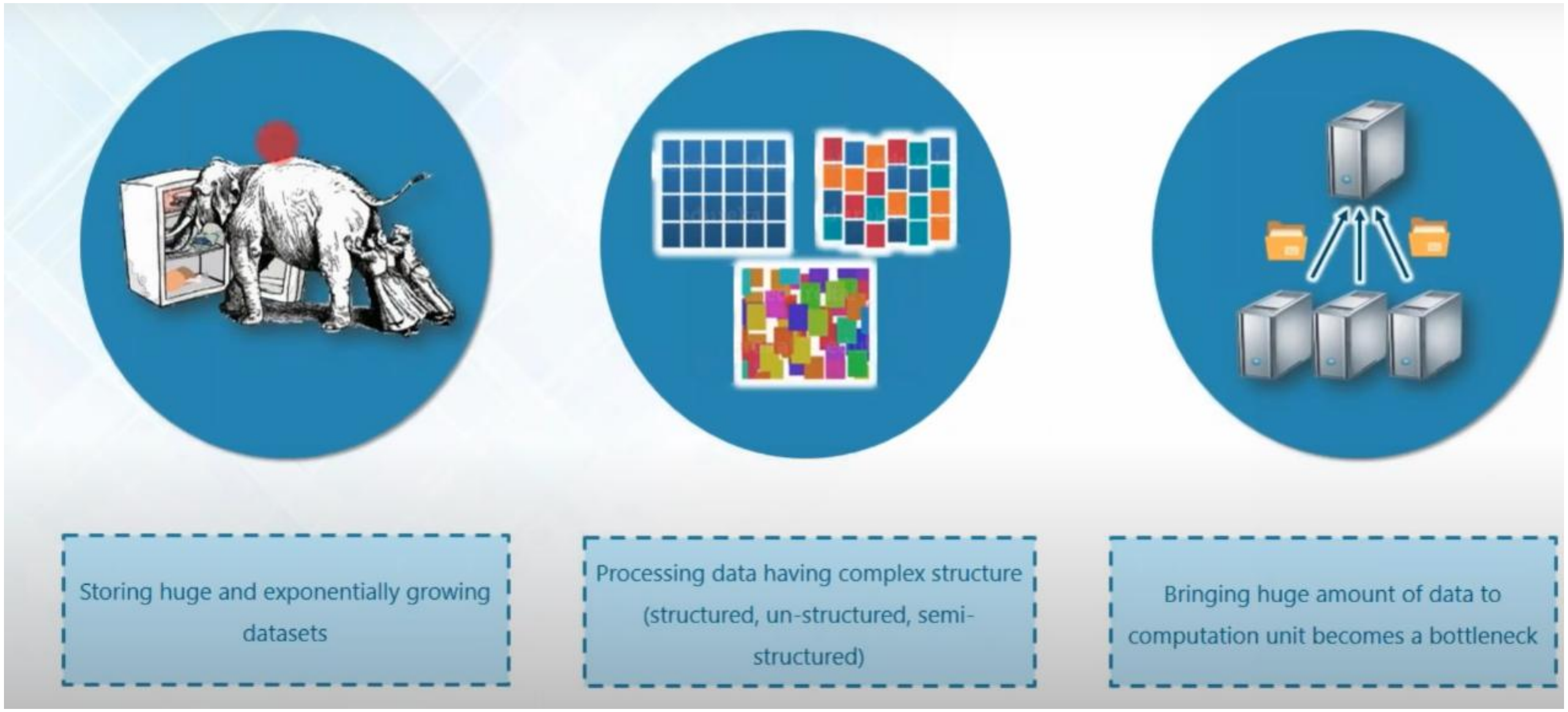
Value

Mechanism to bring the correct meaning out of the data

Veracity

Uncertainty and inconsistencies in the data

## How BigData helps…



Storing huge and exponentially growing datasets

Processing data having complex structure (structured, un-structured, semi-structured)

Bringing huge amount of data to computation unit becomes a bottleneck

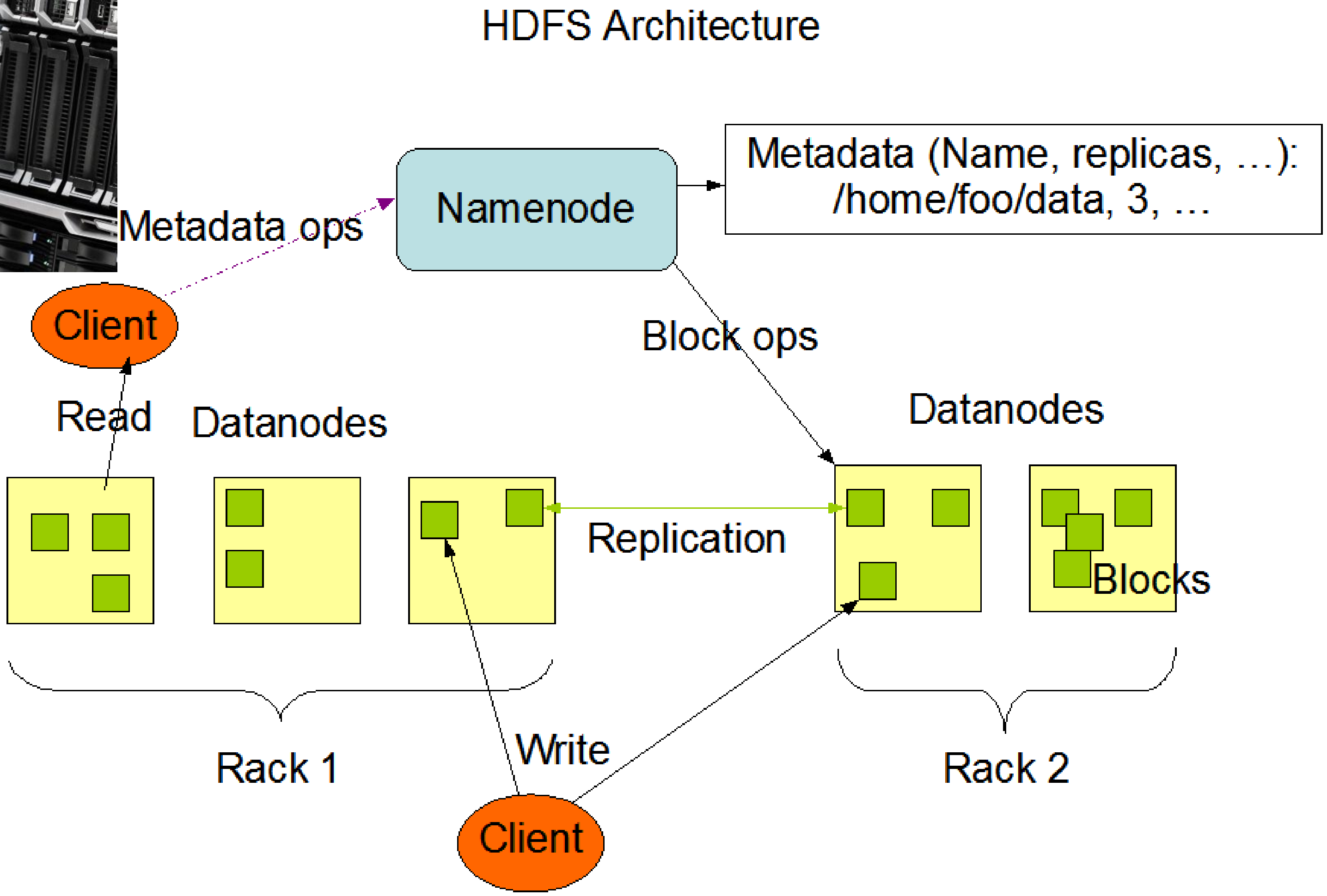
## Goals of HDFS

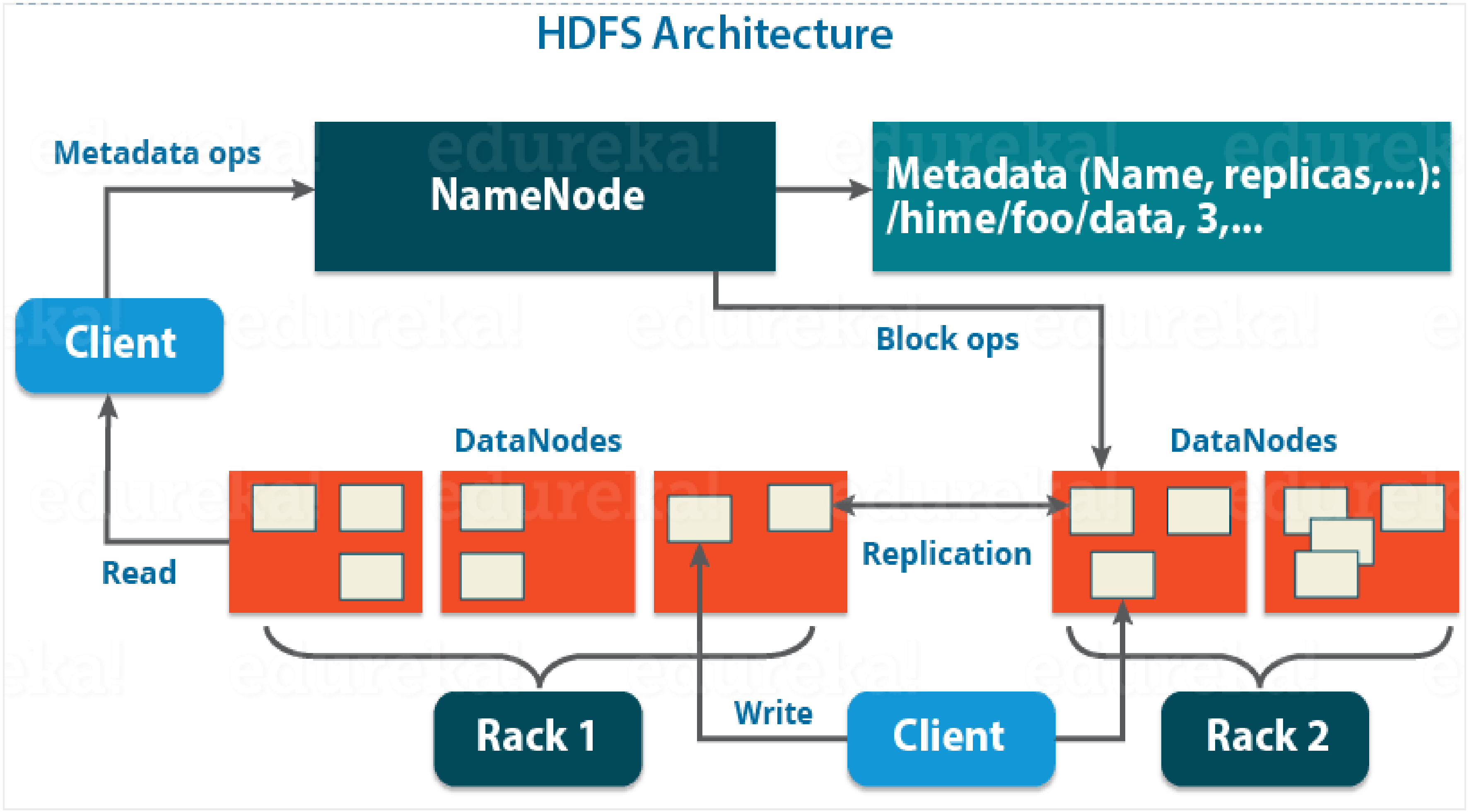
- Very Large Distributed File System
  - 10K nodes, 100 million files, 10PB
- Assumes Commodity Hardware
  - Files are replicated to handle hardware failure
  - Detect failures and recover from them
- Optimized for Batch Processing
  - Data locations exposed so that computations can move to where data resides
  - Provides very high aggregate bandwidth

## Features of Distributed File System

- Single Namespace for entire cluster
- Data Coherency
  - Write-once-read-many access model
  - Client can only append to existing files
- Files are broken up into blocks
  - Typically 64MB block size
  - Each block replicated on multiple DataNodes
- Intelligent Client
  - Client can find location of blocks
  - Client accesses data directly from DataNode

## HDFS Architecture




HDFS Architecture
Metadata ops
NameNode
Metadata (Name, replicas,...): /hime/foo/data, 3,...
Client
Block ops
DataNodes
DataNodes
Read
Replication
Rack 1
Write
Client
Rack 2

## Functions of a NameNode

- Manages File System Namespace
  - Maps a file name to a set of blocks
  - Maps a block to the DataNodes where it resides
- Cluster Configuration Management
- Replication Engine for Blocks

## NameNode Metadata

- Metadata in Memory
  - The entire metadata is in main memory
  - No demand paging of metadata
- Types of metadata
  - List of files
  - List of Blocks for each file
  - List of DataNodes for each block
  - File attributes, e.g. creation time, replication factor
- A Transaction Log
  - Records file creations, file deletions etc

## DataNode

- A Block Server
  - Stores data in the local file system (e.g. ext3)
  - Stores metadata of a block (e.g. CRC)
  - Serves data and metadata to Clients
- Block Report
  - Periodically sends a report of all existing blocks to the NameNode
- Facilitates Pipelining of Data
  - Forwards data to other specified DataNodes

## Block Placement

- Current Strategy
  - One replica on local node
  - Second replica on a remote rack
  - Third replica on same remote rack
  - Additional replicas are randomly placed
- Clients read from nearest replicas
- Would like to make this policy pluggable

## Heartbeats

- DataNodes send hearbeat to the NameNode
  - Once every 3 seconds
- NameNode uses heartbeats to detect DataNode failure

## Replication Engine

- NameNode detects DataNode failures
  - Chooses new DataNodes for new replicas
  - Balances disk usage
  - Balances communication traffic to DataNodes

## Data Correctness in Hadoop

- Use Checksums to validate data
  - Use CRC32
- File Creation
  - Client computes checksum per 512 bytes
  - DataNode stores the checksum
- File access
  - Client retrieves the data and checksum from DataNode
  - If Validation fails, Client tries other replicas

## NameNode Failure

- A single point of failure
- Transaction Log stored in multiple directories
  - A directory on the local file system
  - A directory on a remote file system (NFS/CIFS)

If **NameNode** gets fail the whole **Hadoop** cluster will not work. Actually, there will not any data loss only the cluster work will be shut down, because **NameNode** is only the point of contact to all DataNodes and if the **NameNode** fails all communication will stop.

**To handle the single point of failure, we can use another setup configuration which can backup NameNode metadata. If the primary NameNode will fail our setup can switch to secondary (backup) and no any type to shutdown will happen for Hadoop cluster.**

## Data Pieplining in Hadoop

- Client retrieves a list of DataNodes on which to place replicas of a block
- Client writes block to the first DataNode
- The first DataNode forwards the data to the next node in the Pipeline
- When all replicas are written, the Client moves on to write the next block in file

A **data** pipeline is an arrangement of elements connected in series that is designed to process the **data** in an efficient way.
Then you might have to use MapReduce to process the **data**. To store **data**, you can use SQL or NoSQL database such as HBase. To query the **data** you can use Pig or Hive.

## Job of Secondary NameNode

- Copies FsImage and Transaction Log from Namenode to a temporary directory
- Merges FSImage and Transaction Log into a new FSImage in temporary directory
- Uploads new FSImage to the NameNode
  - Transaction Log on NameNode is purged

## User Interface (Some important commands)

- Commands for HDFS User:
  - hadoop dfs -mkdir /foodir
  - hadoop dfs -cat /foodir/myfile.txt
  - hadoop dfs -rm /foodir/myfile.txt
- Commands for HDFS Administrator
  - hadoop dfsadmin -report
  - hadoop dfsadmin -decommision datanodename