BIG DATA (UE19CS322)

# YET ANOTHER HADOOP

## PROJECT REPORT

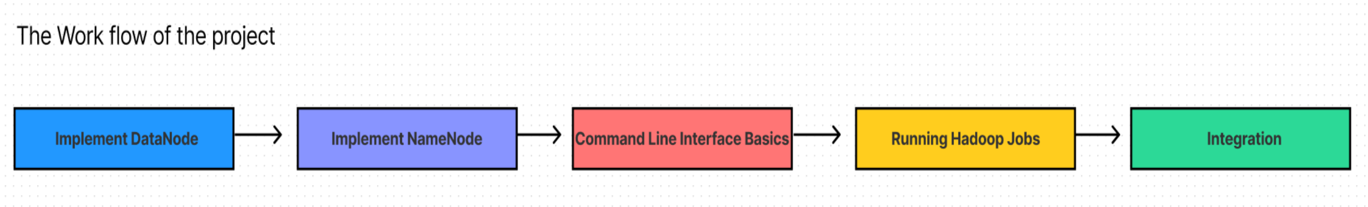
**TeamID: BD_039_096_217_525**

**Team Members:**
Akanksha Akkihal(PES1UG19CS039)
Avanish V Patil(PES1UG19CS096)
Kedar U Shet(PES1UG19CS217)
Surya M N(PES1UG19CS525)

## Introduction

Yet Another Hadoop is a mini-HDFS setup on your system, complete with the architectural structure consisting of Data Nodes and Name Nodes and replication of data across nodes and capable of performing some of the important tasks a distributed file system performs, running HDFS commands as well as scheduling Hadoop jobs.

## Design Details



The above-mentioned workflow gives a high-level design implementation of the project. Each of the implementations is a module in the project.

**Design of each component :**

1) Datanode
   - We have implement data nodes as folders on our system
   - Hash function implemented is a simple **modulus function** which determines the datanode for that particular split of the file
2) Namenode
   - Both primary and secondary namenodes are a single JSON file which contains file name and information about its splits and corresponding data nodes which it has been allocated
   - In case of failure of primary datanode, the secondary datanode is used as a backup through which the primary datanode can be restored.
3) Hadoop Jobs
   - Mappers are implemented on multiple threads and reducer is implemented on a single main thread.

## Surface level implementation details about each unit

**Datanodes**: These are the nodes where the user actually stores data. The user can specify the block size and also the replication factor

**Namenodes**: The name node helps us to access the virtual directories. The primary name nodes  gives the information about the datanode storage. Secondary name node is a backup to the primary name node.

**Command Line Interface**: The CLI is a basic interface for the user to interact with the DFS and run various commands and Hadoop Jobs.

**Hadoop Jobs**: This function helps the user to run the MapReduce Hadoop Job.  We have implemented **multi-threading** to run multiple mappers parallely.

## The reason behind design decisions

The workflow of the project was designed in such a way that all the dependencies and prerequisites are fulfilled (For eg: Once we have implemented Datanodes it gives us a good base for structuring the Namenode).

1) **Datanode :**
    - The hash function is chosen as a mod function for easier split allocation and for easier access of files.
    - The file split will be handled carefully so that in case of a json file a single object will not be broken and split into 2 different datanodes. The reason being it may cause problems to access these objects while running hadoop jobs.
2) **Namenode :**
    - The namenode is implemented as a single json file because it is the best format available to store a file as a key and information about its splits and other metadata as values.
3) **Hadoop Jobs :**
    - In order to run a MapReduce task multiple mappers run on multiple threads to increase performance by achieving parallelism. The output of the mappers is then sorted and given to the reducer which runs on a single main thread
4) **Command Line Interface :**
    - We have written a menu based while loop which asks for user input (which are the commands) until the user exits. This was simple and quick to implement.
5) **Loading DFS :**
    - In case the file split is deleted due to some reason, then the data will be replicated if the replica exists else the entire file will be removed.

6) **Sync period and checkpoints:**

   - If the primary node goes down the secondary namenode sends a heartbeat to the primary namenode and if it detects a failure the primary namenode will be restored

## The takeaway from the project

We learnt a lot about the internal functioning of HDFS like working of Datanodes, Namenodes, Splitting of files and the way the blocks are assigned to different datanodes. We also learnt a lot about file systems on python and various OS functions. Overall this was a  great and practical hands-on project to work on.