PROJECT TITLE: YAH -YET ANOTHER HADOOP

Design details:

YAH is meant to be 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. We ensured that there was a proper communication between datanode and namenode. In a config.json file we have added configuration details of the DFS. The setup process reads in this config file and creates the DFS based on the configuration provided by us. Data is only stored in the Data Nodes in blocks. The Name Node is in charge of mapping file system operations performed in this directory to files present in the Data Nodes. This path is used as a root directory to manipulate the distributed file system.

Surface level implementation details about each unit:

1) Namenode.py
   Here we have imported the .config file we had created and tried to implement the parent directory if it does not exist. And also created NameNodeServer .We then run this on the localhost port number 18812.

2) DataNode.py
   We have created a DataNodeServer and tried to run on the portal 8888. Note that this gets executed only if the Namedone run properly.

3) Client_code.py
   Here, we have created 4 options. They are
   1. Make directory
   2. Remove directory
   3. Display the directory structure
   4. Exit
   Basic if else loop is used to implement this. It is also noted here that it runs only if the namenode gets executed.

4) Setup.py
   This is a new config_setup python file which automatically creates a new dfs setup config. It asks the user if you have a config file created before. If the user presses y it gets created automatically and the number of nodes gets incremented. If n is pressed it won't create a new one.

Reason behind design decision:

We had strong basic skills in hadoop hdfs architecture which made us take up this project. And it was easy for us since we were familiar with the bash, config files while installing the hadoop. We also got more exposure to possess finer control over your data blocks and their replication. We also chose this as it allows running Hadoop like jobs to break down computationally heavy tasks into smaller distributed tasks.

Take Away from the project:

We have created a mini HDFS. This is useful for the basic needs such as create directory, remove directory, To display the directories if exists, and to exit from it.

This project also has a feature of automatically creating dfs config files. Which helps us to get to know about the one we have and the number of files.

Name node Server and the Data node Server helps us to connect to the portal. We can also check the log files in the name node. The checkpoints are implemented. The working functions were made.