Big Data Project report

Team ID - BD_191_264_438_461

YAH – Yet Another Hadoop

This project is an attempt to simulate a miniature HDFS capable of performing some of the important tasks a distributed file system performs, running HDFS commands as well as scheduling Hadoop jobs.

Design and implementation details

Creating/Loading DFS : DFS is created/Loaded based on the given configuration file. If DFS does not exist then a new DFS with Namenode, Datanodes, Secondary Namenode is created. Otherwise previously created DFS is loaded.

Namenode tracks information related to file to block mapping, location of each block and its replica and the file system directories.

Secondary datanode is used to backup the information of Namenode.

Datanode consists of all the file blocks created by the user.

Command line interface : A command line interface is created using the argparse library. User can execute commands like put, ls, cat, rm, mkdir, rmdir and mapreduce. Namenodes and Datanodes are appropriately updated after execution of these commands.

Block distribution : When a file is submitted to the DFS it is divided into multiple blocks and replicated. The file blocks are distributed to the datanodes in a round-robin fashion such that each replica goes to a different datanode.

Fault tolerance : Namenode periodically sends a heartbeat signal to each of the Datanodes to check for the existence of the file blocks. If a file block or a Datanode is missing then they are regenerated using the replicas.

Namenode failure : If Namenode fails all the data that is backed up in Secondary name node is used to bring back the Namenode.

Running hadoop jobs : This functionality is implemented using the subprocess library. cat command is used to get the input file details from the DFS. The output is passed to the mapper submitted by the user. Finally the output of mapper is sorted and submitted to the reducer. The reducer output is temporarily stored in a file which is used by the put command to send the output back the DFS.

Implementation Files

setup.py : Used to create the DFS

load.py : Loads the DFS

commands.py : functions for all commands namely put, ls, cat, rm, mkdir, rmdir and mapreduce.

heartbeat.py : Periodically checks datanodes and recreates in case of failure. (Fault tolerence)

zookeeper.py : Periodically checks for namenode failure and takes suitable action.

utilities.py : Utility functions like filesplit, updating json etc

main.py : Execution of all functionalities.

Takeaway from the project

1. Good understanding of Hadoop distributed file system and its architecture.
2. Team work