- MapReduce has two phase: map and reduce. Map will split the input source and assign them into key and value pairs. And reduce phase will take the output from map phase, and apply for the reduce function. The reduce function will reduce result for (K2, V2) tuples. For example, in MaxTemperature example, map phase will take all data and pair them as (year, temperature) tuples. Then in the reduce phase, year will be taken as a key and find the max temperature associated with this year.
- Read operation:
    - client opens the required file by calling open() on Filesystem
    - DistributedFileSystem interacts with Namenode to get the block location of file to be read
    - DistrubutedFileSystem returns an FSDataInpuStream
    - Blocks are read in order
    - When finishing reading, it calls Close() on FSDataInputStream to close connection

  Write operation

    a. client creates the file by calling create() method on DistrubutedFileSystem
    b. DistributedFileSystem makes an RPC call to namenode to create a new file in the filesystem's namespace
    c. As the client writes data, DFSOutputStream splits it into packets
    d. The second datanode stores the packet and forwards it to the third and last datanode in the pipline
    e. DFSOutputStream maintains an internal queue of packets that are waiting to be acknowledged by datanodes
    f. When finished writing data, it call close() on the stream
    g. It flushes all the remaining packets to the datanode pipeline and waits for acknowledgments.
- Shell commands
    - setrep: changes the replication factor of a file. -R flag is accepted for backwards compatibility. -w flag requests that the command wait for the replication to complete.

      hadoop fs -setrep [-R] [-w] <numReplicas> <path>

      chown: Change the owner of files. The use must be a super-user. -R option will make the change recursively through the directory

      hadoop fs -chown [-R] [OWNER][:[GROUP]] URI [URI]

      touchz: create a file of zero length

      hadoop fs -touchz URI [URI ...]

      mv: move files from source to destination

      hadoop fs -mv URI [URI ...] <dest>

      mkdir: take path uri's as argument and creates directories

      hadoop fs -mkdir [-p] <paths>