Caleb Post
Lab 1 : Introductuion to MapReduce & HDFS

Question 1: Explain the MapReduce concept in your own words using a simple example.
The map reduce concept is in essence decoupling the processing of the data from the data itself. You write a method which has no knowledge of the amount of data it is running on, it could be one line, it could be terabytes of data. This map function breaks the data apart and processes it into whatever form you want. The power of this is that Hadoop can then break the data into smaller sub problems, distributing them among the nodes connected, which can then in turn distributed if needed and the network is set up that way. The answers are passed back to the master node then, and the reduce step takes place. The reduce step is the combination of all the smaller sub problems into a final output, whatever that output may be.

The power of this approach is that all the map steps can be done in parallel, breaking large computations down into relatively small processes on many machines at once. The data from this then needs to be combined into the final answer, the reduce step.

An example of this would be calculating the maximum temperature from a weather satellite network. This satellite network is broadcasting gigabytes of temperature data to the NSF for study each day. This data is added in text files to their Hadoop cluster.

When the max temperature job is run, the map function is run on each node containing pertinent temperature files. We don't know how many times this will run, but it will be run over all the temperature data.

The intermediate temperatures are then passed into a reducer. The reducer pulls the maximum temperature from all the processed temperatures and writes it as output.

Question 3: Explain in your own words, the data flow during a read operation on HDFS and the data flow during a write operation on HDFS.

During a read operation, data flow is between the client and the namenode first, and then the client and the data nodes. The client requests the block locations for the data from the NameNode. It then receives these locations. It then requests specific blocks from the data nodes with regards to both availability and proximity. It will get the blocks from the best name node that it can.

During a write operation, the client first notifies the name node of the new file that is going to be created. If there are no conflicts, it then writes to a data output stream. This stream breaks the data into packets that will be written to the various data nodes. Packets are sent from a queue to the first data node where the data will be replicated. That node will then write the packet and forward it to the second data node, and so on. An acknowledgement is required at each step that the packet was written successfully. Once all the packets have been acked, the locations of the blocks are then registered with the name node.

In a failure by a datanode to write, the system notifies the namenode of the error and removes the failed datanode from the pipeline to be written to. It will then see that the data is underreplicated and add another node to the pipeline.

Question 4: Explain the following HDFS Shell commands with an example. Each command is worth a point.

1. setrep
    a. setrep is used to change the replication factor of a file or files within a directory
    b. example: *hadoop fs –setrep –w 3 –R /user/Hadoop/dir1* to set the replication factor to 3 on all files in the specified directory
2. chown
    a. change the owner of a file(s).
    b. example: *Hadoop fs –chown caleb:users /user/Hadoop/dir1/example.txt*
3. touchz
    a. creates a file of length zero.
    b. Example: *hadoop fs –touchz /user/Haddop/emptyFile.txt*
4. mv
    a. moves the file from source to destination
    b. Example: *hadoop fs –mv /user/Hadoop/file1 /user/Hadoop/file2*
5. mkdir
    a. creates the directories that comprise the path specified. Works like unix mkdir –p because it creates all the parent directories as well.
    b. Example: *hadoop fs –mkdir /user/hadoop/nest/more/dir1*