**Ques 1.) Explain what is checksum and the importance of checksum and how hadoop performs checksum?**

Checksum is a calculated value that is used to determine the integrity of data. Checksum serves as a unique identifier for the data (a file, a text string, or a hexadecimal string). If the data changes then so does the checksum value. This makes it easy to verify the integrity of the data.

To test data integrity, the sender of the data calculates checksum value by taking the sum of the binary data transmitted. When receiving the data, the receiver can perform the same calculation on the data and compare it with the checksum value provided by the sender. If the two values match, the receiver has a high degree of confidence that the data was received correctly.

Checksum value is also called hash value. The data that is calculated can be a file, a text string, or a hexadecimal string.

**Use checksum values**

**First,** checksum value can be used to check data integrity when data is sent through telecommunication networks such as Internet.

**Second,** checksum value can be used to check data integrity of stored data to see if the data has been modified or changed in any way over time. Data can be modified in many ways. It may be infected by viruses, packet loss when transferring through networks, accidental or intentional human changes of data, or anything else.

**Third,** checksum values can be used to verify data burned to CDROM, CD-R (Compact Disc-Recordable), OR DVD, DVD-R.

**Hadoop perform checksum by**

1) A separate checksum is created for every dfs.bytes-per-checksum bytes of data.

2) The default is 512 bytes, and because a CRC-32C checksum is 4 bytes long. The storage overhead is less than 1%.

Input checksum



3) Datanodes are responsible for verifying the data they receive before storing the data and its checksum.

4) When the clients read data from datanodes, they verify checksums as well, comparing them with the ones stored at the datanodes.

5) -get command does the checksum verification during the data read.

6) -copyFromLocal doesn’t perform checksum during data read.

7)-ignoreCrcoption with the -get is equivalent to –copyToLocal command.

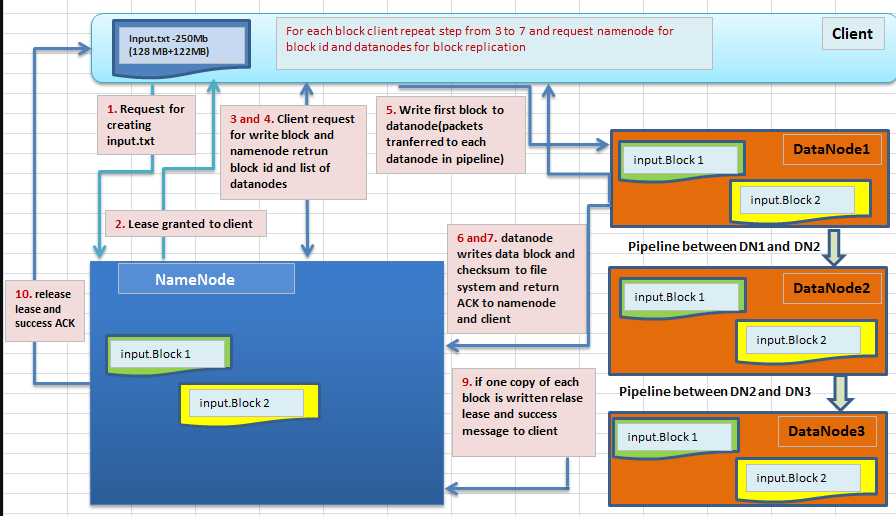
8) Disabling checksum verification is useful if we have a corrupt file that we want to inspect so that we can decide what to do with it.

**Q2. Explain the anatomy of file write to HDFS?**

Ans)

In hadoop cluster, data is stored in file(create new or append in existing one) and file is break into multiple blocks.Once data is written in file it cannot be deleted only append new data is possible, HDFS follows single-writer, multiple-reader model. Lets understand how hadoop uses this model and perform write operation.

1. When HDFS client request Namenode to create a new file(say input.txt), namenode checks permission for that and create appropriate inode.
2. The HDFS client that creates/open file for writing is granted lease for the file and it is locked for that client and no other client has write access on it(Remember single writer,multiple reader model). Client has to renew lease before it expires by sending light weight message.Duration of lease is bounded by soft limit and hard limit.  
   Soft limit:- time duration before which if client did not renewed lease, other client can take over write access form this client.  
   Hard limit:- time duration(1 hour) after which write access will be revoked automatically, if lease is not renewed.
3. From namenode HDFS client requests for block where file will be stored (Remember in HDFS file is stored in blocks).



Namenode returns unique block id and list of datanodes for block replicas to be stored.

1. Client write first block on datanode (block id which was returned by namenode) and passes list of data-nodes(provided by namenode for block replication) to that datanode where first block was written.Along with actual data client also send checksum to datanode for that block.  
   Client send data block in form of collection of packets to first datanode and datanode form a pipeline with list of datanodes which minimizes the total network distance from the client to the last datanode. Consider following diagram to understand how block splits into packets and one block is written to multiple datanodes:
2. First pipeline is set-up, then data streaming is carried out by sending packets and for each packet acknowledgement is send back to client and finally pipeline connection is tear down.
3. The datanodes writes data block in file system and also stores checksum metadata in separate file,which is used while reading to verify data is valid or corrupted.
4. Datanode acknowledges namenode and client about block persisted to file system.
5. After cycle for one block completes,again client ask namenode for block id and datanodes and above process (4 to 7) repeats for other .
6. After all blocks have be consumed, namenode checks if at least one copy of each block has been written.
7. If above check is successful,then it release lease and send success message to client.

**Q3. Explain how HDFS handles failures during file write?**

Ans)

1. The pipeline is closed and any packets in the ackqueue are added to the front of the data queue.

2. The current block on the good data nodes is given a new identity, which is communicated to the name node

3. The failed data node is removed from the pipeline, and a new pipeline is constructed from the two good datanodes.

4. The remainder of the block’s data is written to the good data nodes in the pipeline.

5. The name node notices that the block is under-replicated, and it arranges for a further replica to be created on another node.

6. As long as dfs.namenode.replication.minreplicas (which defaults to 1) are written, the write will succeed.

7. The block will be asynchronously replicated across the cluster until its target replication factor is reached (dfs.replication, which defaults to 3).