**Cloud Computing for Data Analysis**

**VIDEO CASE 02: Hadoop HDFS**

|  |
| --- |
| **Submitted by:** Abdullah Al Raqibul Islam  **UNCC ID:** 801151189 |

Watch following videos:

**Video 1**: <https://youtu.be/cC6sS80sZYQ>

**Video 2:** <https://youtu.be/yDV0EE9DCJ0>

**Video 3:** <https://youtu.be/nbOagGnIMiY>

Video 1 introduces you to HDFS

Video 2 describes about Name Node in HDFS

Video 3 provides knowledge about file read and write operations in HDFS

**Video Case Questions:**

1. Give some description about data nodes and name node

|  |
| --- |
| Data Node: In Hadoop echo-system, Data Node is responsible to store the data in the local file system. It stores metadata of the data block and serve both the data and metadata requested by the client.  Name Node: In Hadoop echo-system, Name Node manages filesystem namespace as well as cluster configuration management. It is considered as the master node and stores the file status distributed across the cluster. It maps both the file name to a set of blocks and a block to the set of data-nodes where it will be places. It keeps track of the complete file system and replications of the file blocks. It works smartly so that it can handle failure of a data node (i.e. failure of replica of a file or the whole rack). |

1. What is the main purpose of secondary name node?

|  |
| --- |
| As name node is considered as the master node and responsible for lots of crucial tasks (i.e. data replication, management of data node failure etc.), a single name node will become the single point of failure for the whole system. That’s why we need secondary name node to assist the name node to release the resources (i.e. main memory). Secondary name node doesn’t function like name node. Rather, it’s main purpose is to combine the name space image and edit logs so that the main memory of name node will not fills up by the ever increasing edit logs. It also creates the checkpoint of namespace image and edit log merge together and write it to a file so that name node can release the main memory till the checkpoint. |

1. What are all the steps followed by HDFS for write operation?

|  |
| --- |
| 1. Client communicate with the name node with a write request 2. Name node perform various check (i.e. if the file exists or not, client has the correct permission or not for the activity) 3. If the check phase pass, the name node will respond the client with the list of data nodes where the data will be copied 4. The client then connects to the first data node and ask it to make a pipeline for the subsequent data nodes 5. Data nodes will send the acknowledgement to the client as they successfully copy the blocks 6. Steps 3-5 will be repeated until all the chunks of the file has been written to the HDFS. 7. After that, the client will send a completion message to the name node 8. In case of data node failure, the data is written on the remaining other data nodes. Name node notice the under replication and arrange for the replication. Same steps will be followed for the case of multiple node failure. The data is necessary to be written to at least one node and the under replicated chunks will be taken care by the name node. |

1. Explain the steps that Hadoop follows for reading the data during a data node failure.

|  |
| --- |
| Bandwidth is the key concern for the read operation. There is a distance concept in HDFS based on the bandwidth. 2 data blocks referred from the same data node have 0 distance. If the blocks stored on the same rack but different data node, then the distance is counted as 2. The distance is considered as 4 and 6 in the case of different racks and different data center respectively. Now come to the read operation steps   1. HDFS client sends a read request to the name node. 2. In response, the name node provides a list of data nodes containing the first few blocks. Name node provide the data node list with the increasing distance of the data nodes. 3. Then the client will send the first data nodes and read the blocks one by one. 4. In case of a data node failure,    1. If the data is corrupted, the client will forward the read request to the next data nodes.    2. If it is a node failure case, the HDFS client will keep this information to remove the further appearance of that data node in the next responses it gets from the name node (for other blocks). |