SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name – Big Data Analytics
III Year / V Semester

Unit 3 – DATA ANALYTICAL FRAMEWORKS

Topic - HDFS – Block Replication, Read, Write

HDFS - Block Replication

Block and Replication

- Blocks are the nothing but the smallest continuous location on your hard drive where data is stored. In general, in any of the File System, you store the data as a collection of blocks.

- Generally the user data is stored in the files of HDFS. The file in a file system will bedivided into one or more segments and/or stored in individual data nodes.

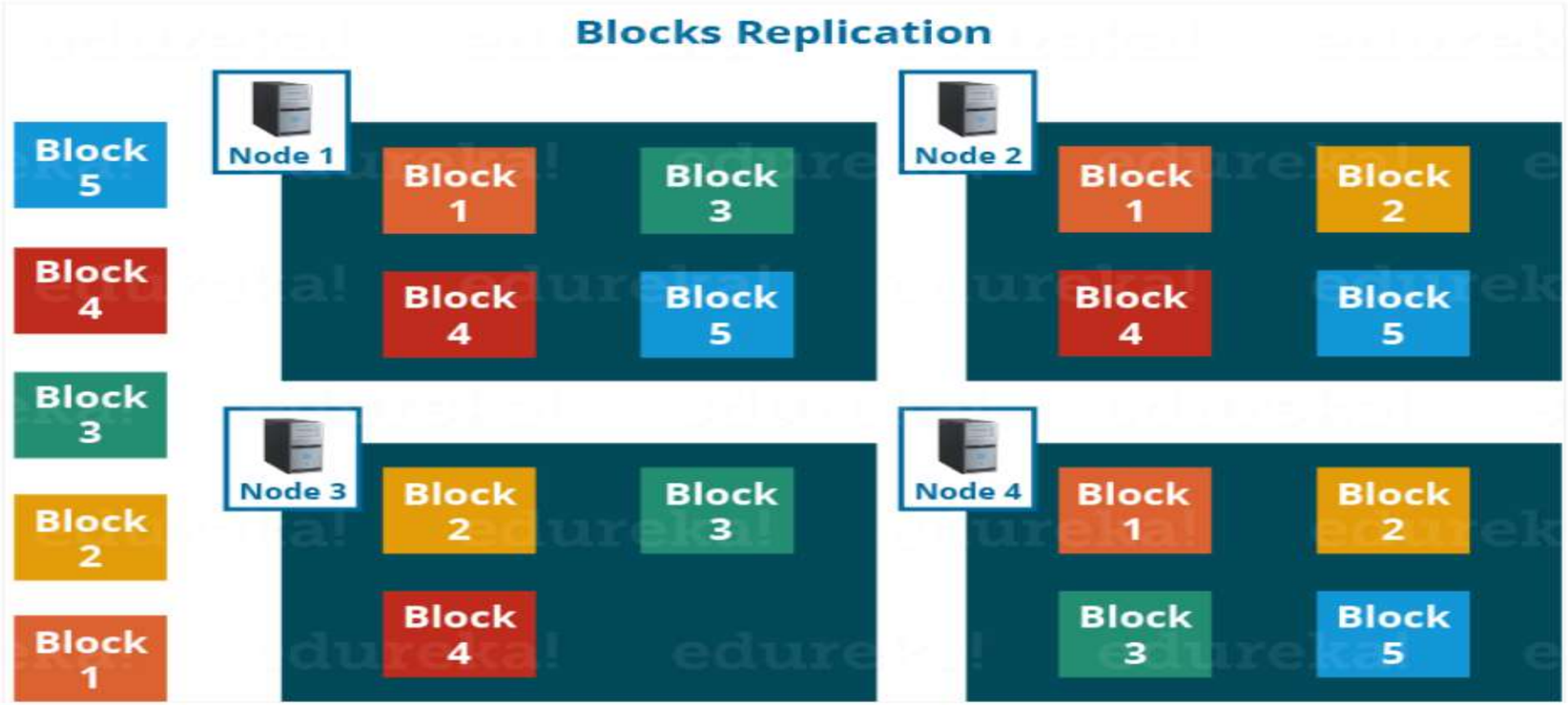
- These file segments are called as blocks. In other words, the minimum amount of datathat HDFS can read or write is called a Block.

- The default block size in Hadoop version 1 is 64MB, but it can be increased as per the need to change inHDFS configuration. Default block size in Hadoop version 2 is 128 MB.

HDFS - Block Replication

Replication Management:

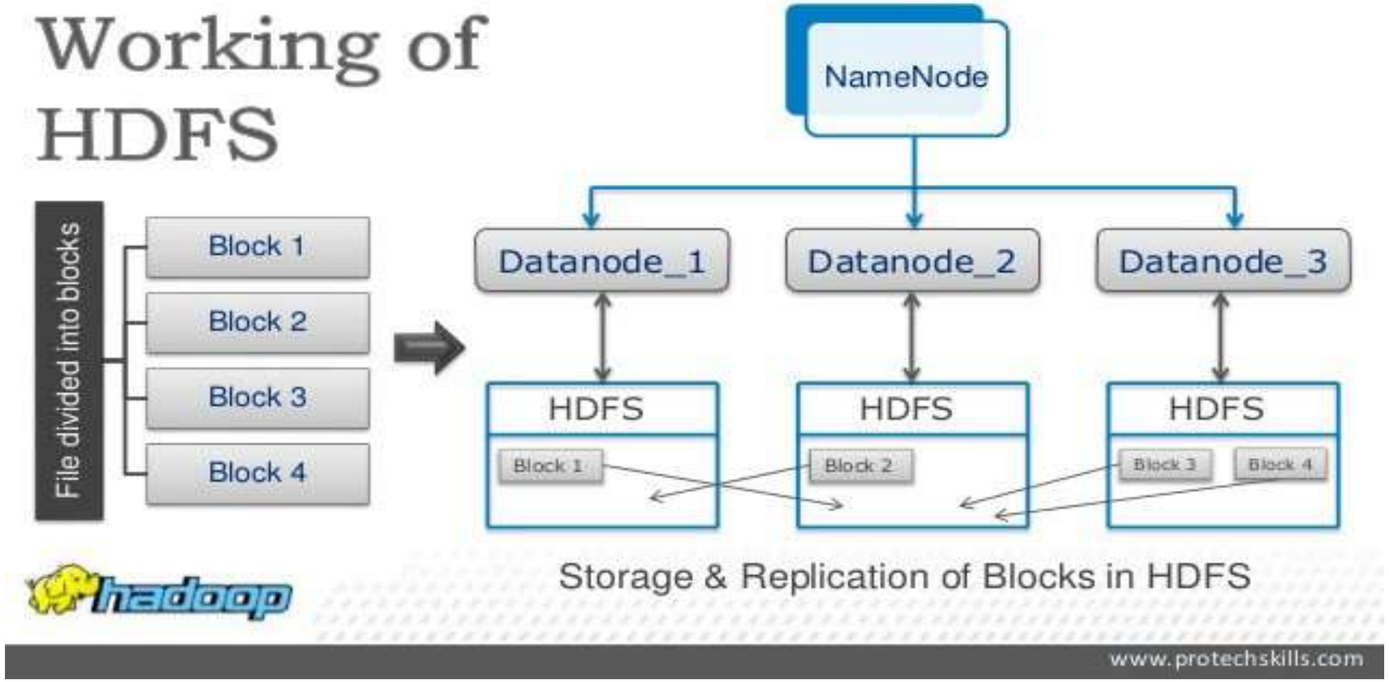
- HDFS provides a reliable way to store huge data in a distributed environment as data blocks. The blocks are also replicated to provide fault tolerance.
- The default replication factor is 3 which is again configurable.



HDFS - Block Replication

Monitoring:

- There is a continuous “heartbeat” communication between the data nodes to the name node.
- If a data node’s heartbeat is not heard by the name node, the data node is considered to have failed and is no longer available.
- In this case, a replica is employed to replace the failed node, and a change is made to the replication scheme.



HDFS – Write Architecture

HDFS Read/ Write Architecture:

- HDFS follows Write Once – Read Many Philosophy. So, you can't edit files already stored in HDFS. But, you can append new data by re-opening the file.

- HDFS Write Architecture:
  Assume that the system block size is configured for 128 MB (default). So, the client will be dividing the file "example.txt" into 2 blocks – one of 128 MB (Block A) and the other of 120 MB (block B).

Now, the following protocol will be followed whenever the data is written into HDFS:

- At first, the HDFS client will reach out to the NameNode for a Write Request against the two blocks, say, Block A & Block B.

- The NameNode will then grant the client the write permission and will provide the IP addresses of the DataNodes where the file blocks will be copied eventually.

HDFS - Write Architecture

- The selection of IP addresses of DataNodes is purely randomized based on availability, replication factor and rack awareness that we have discussed earlier.

- Let's say the replication factor is set to default i.e. 3. Therefore, for each block the NameNode will be providing the client a list of (3) IP addresses of DataNodes. The list will be unique for each block.

- Suppose, the NameNode provided following lists of IP addresses to the client:
  - For Block A, list A = {IP of DataNode 1, IP of DataNode 4, IP of DataNode 6}
  - For Block B, set B = {IP of DataNode 3, IP of DataNode 7, IP of DataNode 9}

- Each block will be copied in three different DataNodes to maintain the replication factor consistent throughout the cluster.

- Now the whole data copy process will happen in three stages:
  - Set up of Pipeline
  - Data streaming and replication
  - Shutdown of Pipeline (Acknowledgement stage)

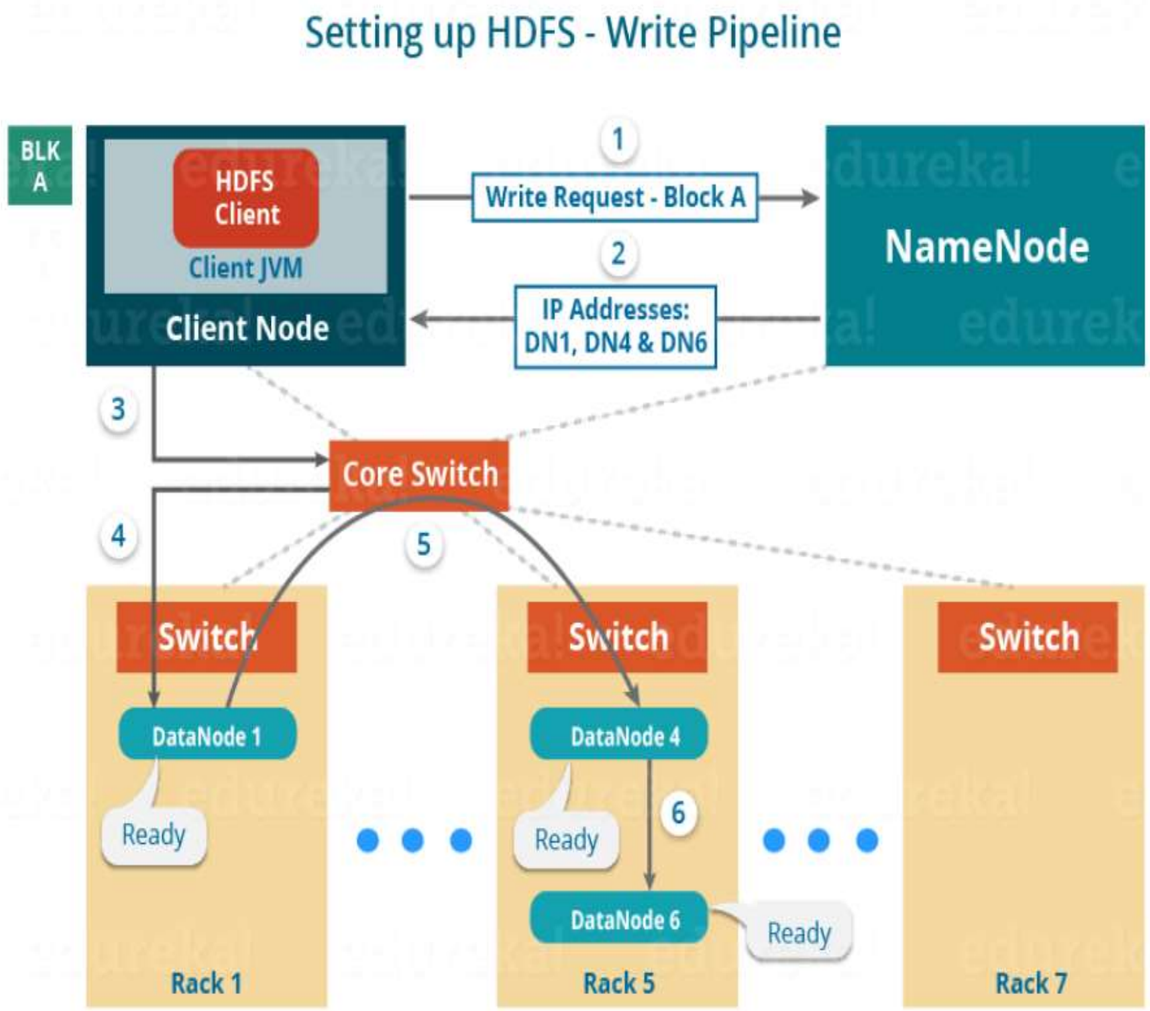
HDFS - Write Architecture

1. Set up of Pipeline:

- Before writing the blocks, the client confirms whether the DataNodes, present in each of the list of IPs, are ready to receive the data or not.

- In doing so, the client creates a pipeline for each of the blocks by connecting the individual DataNodes in the respective list for that block. Let us consider Block A.

- The list of DataNodes provided by the NameNode is:For Block A, list A = {IP of DataNode 1, IP of DataNode 4, IP of DataNode 6}.



HDFS - Write Architecture

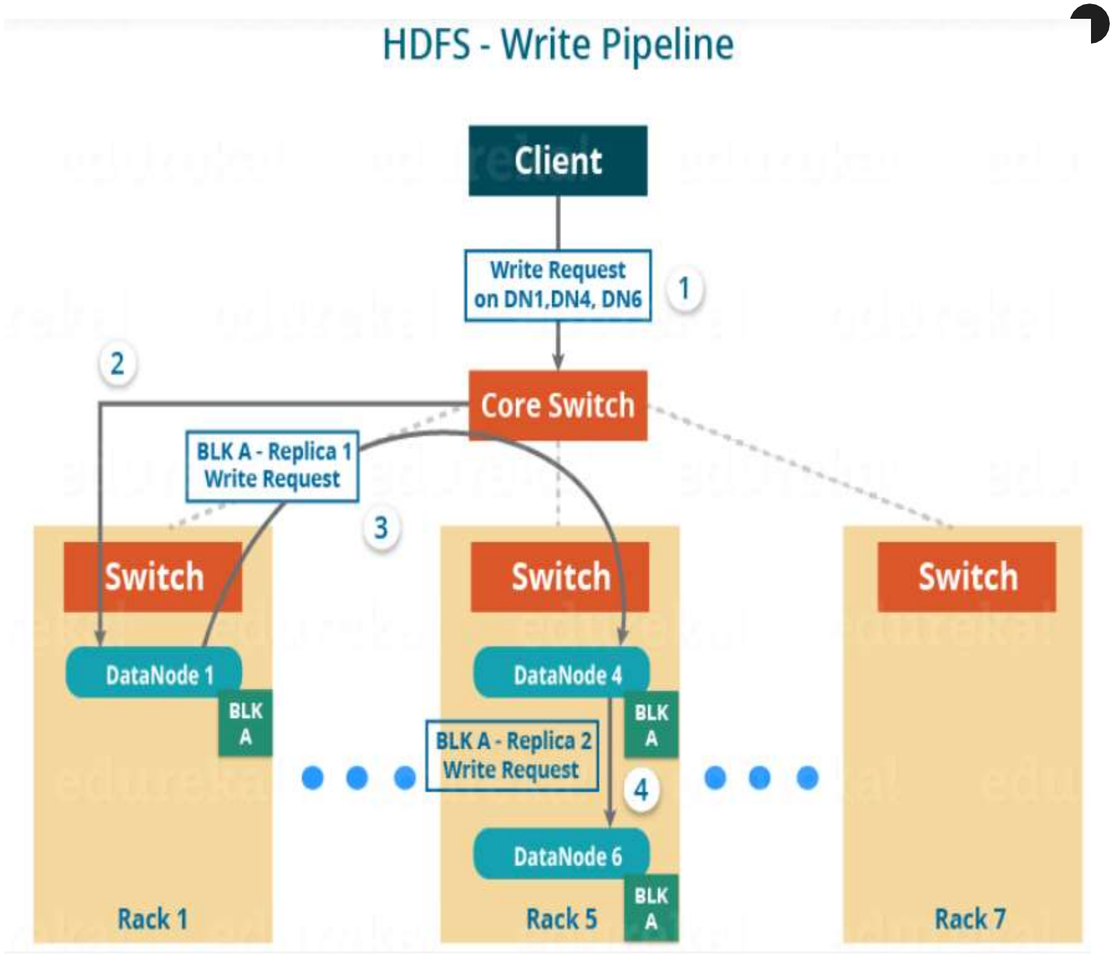
So, for block A, the client will be performing the following steps to create a pipeline:

- The client will choose the first DataNode in the list (DataNode IPs for Block A) which is DataNode 1 and will establish a TCP/IP connection.
- The client will inform DataNode 1 to be ready to receive the block. It will also provide the IPs of next two DataNodes (4 and 6) to the DataNode 1 where the block is supposed to be replicated.
- The DataNode 1 will connect to DataNode 4. The DataNode 1 will inform DataNode 4 to be ready to receive the block and will give it the IP of DataNode 6. Then, DataNode 4 will tell DataNode 6 to be ready for receiving the data.
- Next, the acknowledgement of readiness will follow the reverse sequence, i.e. From the DataNode 6 to 4 and then to 1.
- At last DataNode 1 will inform the client that all the DataNodes are ready and a pipeline will be formed between the client, DataNode 1, 4 and 6.
- Now pipeline set up is complete and the client will finally begin the data copy or streaming process.

HDFS - Write Architecture

2. Data Streaming:

- As the pipeline has been created, the client will push the data into the pipeline.
- Now, don't forget that in HDFS, data is replicated based on replication factor.
- So, here Block A will be stored to three DataNodes as the assumed replication factor is 3. Moving ahead, the client will copy the block (A) to DataNode 1 only.
- The replication is always done by DataNodes sequentially.



HDFS - Write Architecture

So, the following steps will take place during replication:

- Once the block has been written to DataNode 1 by the client, DataNode 1 will connect to DataNode 4.
- Then, DataNode 1 will push the block in the pipeline and data will be copied to DataNode 4.
- Again, DataNode 4 will connect to DataNode 6 and will copy the last replica of the block.

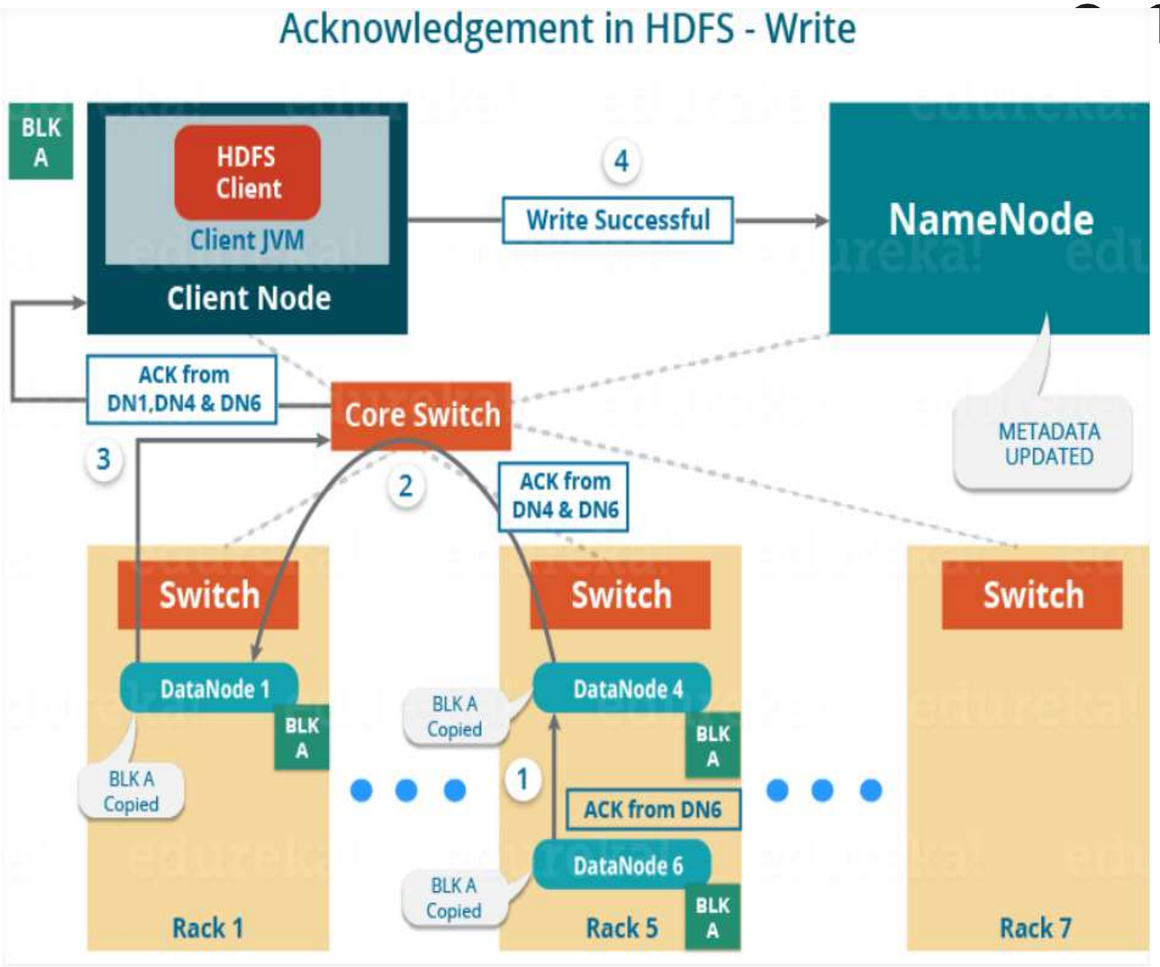
3. Shutdown of Pipeline or Acknowledgement stage:

- Once the block has been copied into all the three DataNodes, a series of acknowledgements will take place to ensure the client and NameNode that the data has been written successfully. Then, the client will finally close the pipeline to end the TCP session.

- As shown in the figure below, the acknowledgement happens in the reverse sequence i.e. from DataNode 6 to 4 and then to 1. Finally, the DataNode 1 will push three acknowledgements (including its own) into the pipeline and send it to the client. The client will inform NameNode that data has been written successfully. The NameNode will update its metadata and the client will shut down the pipeline.

HDFS - Write Architecture

- Similarly, Block B will also be copied into the DataNodes in parallel with Block A. So, the following things are to be noticed here:
- The client will copy Block A and Block B to the first DataNode simultaneously.
- Therefore, in our case, two pipelines will be formed for each of the block and all the process discussed above will happen in parallel in these two pipelines.
- The client writes the block into the first DataNode and then the DataNodes will be replicating the block sequentially.

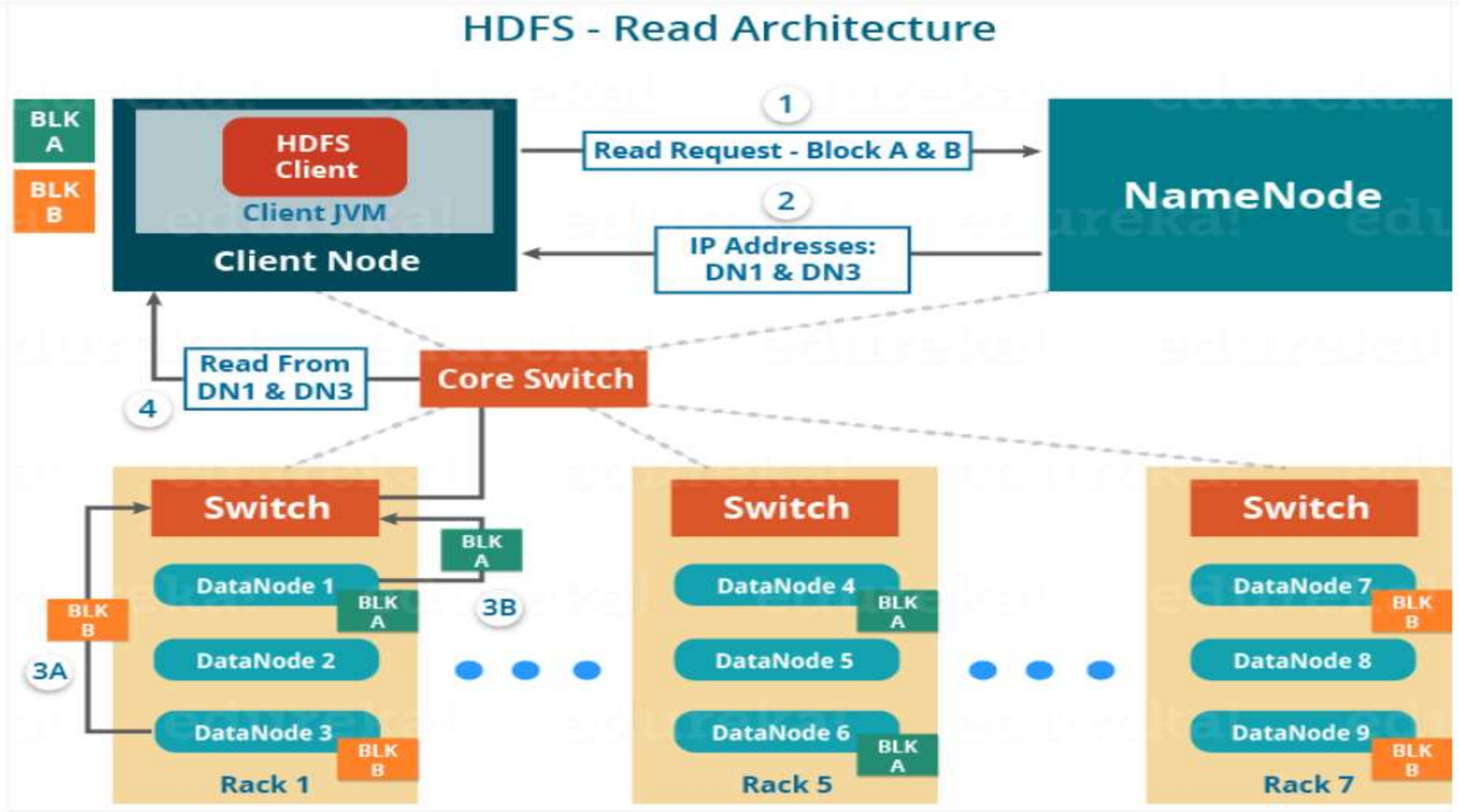


HDFS - Read Architecture

HDFS Read Architecture:

- HDFS Read architecture is comparatively easy to understand. Let's take the above example again where the HDFS client wants to read the file "example.txt" now.Now, following steps will be taking place while reading the file:
- The client will reach out to NameNode asking for the block metadata for the file "example.txt".
- The NameNode will return the list of DataNodes where each block (Block A and B) are stored.
- After that client, will connect to the DataNodes where the blocks are stored.
- The client starts reading data parallel from the DataNodes (Block A from DataNode 1 and Block B from DataNode 3).
- Once the client gets all the required file blocks, it will combine these blocks to form a file.
- While serving read request of the client, HDFS selects the replica which is closest to the client. This reduces the read latency and the bandwidth consumption. Therefore, that replica is selected which resides on the same rack as the reader node, if possible.

HDFS - Read Architecture



THANK YOU