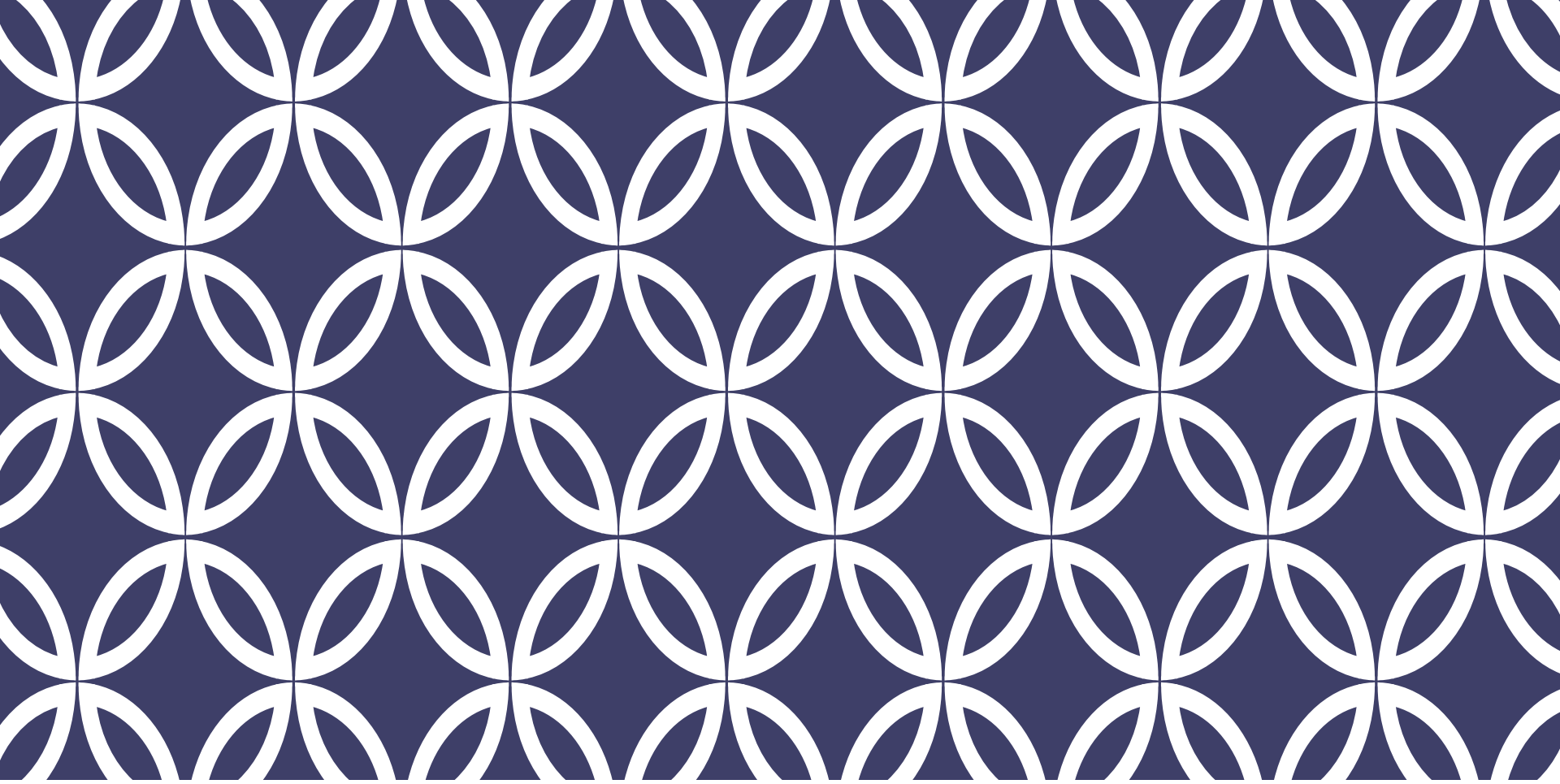
ACADGILD

Presents

Introduction to Big Data and Hadoop2.x



Session 3–HDFS INTERNALS

Agenda – HDFS INTERNALS

- **Hadoop Distributed File System (HDFS):**
  - Introduction
  - Design of HDFS

- **Data Model:**
  - HDFS Data Flow
  - Blocks in HDFS
  - HDFS high level architecture
  - Processing constraint with Hadoop physical block
  - Processing on Input Split.
  - Relation between Hadoop Block and Split
  - HDFS File-Write

Introduction to Hadoop Distributed File System(HDFS)

HDFS is:

- The first component of Hadoop.
- Based on the Google File System (GFS).
- Divides huge data into files and distributes them among the nodes of the cluster.
- One of the nodes in the cluster is set as the Master Node.

HDFS has been designed keeping in view the following features.

- ✓ **Very large files: F**iles that are megabytes, gigabytes, terabytes or petabytes of size.
- ✓ **Streaming data access:** HDFS is built around the idea that data is written once but read many times. A dataset is copied from source and then analysis is done on that dataset over time.
- ✓ **Commodity hardware:** Hadoop does not require expensive, highly reliable hardware as it is designed to run on clusters of commodity hardware.

HDFS – Data Flow

MapReduce job contains:

1. Input data.
2. MapReduce program
3. Configuration information.

- Hadoop divides input into fixed size pieces called **Input Splits** or **Splits.**
- Hadoop creates one map task for each split which runs the user defined Map function for each record in the split.
- For most jobs a good split size tends to be the size of an HDFS block which is **128 MB or 64 MB** by default.
- Map tasks write their output to local disk which is further processed by Reduce task to produce the final output.

Blocks in HDFS

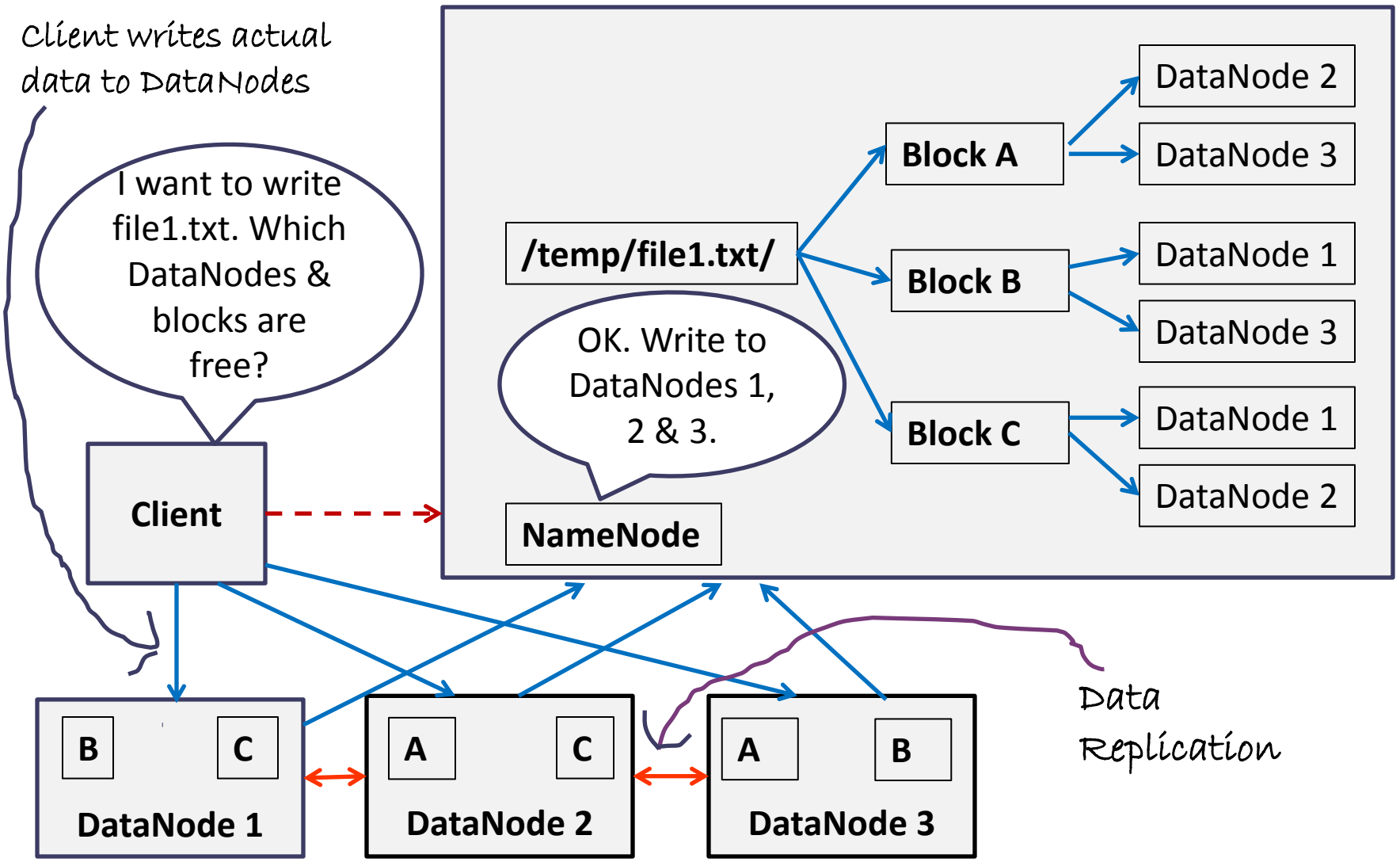
- Files are stored in the format of blocks and the default block size is 128 MB and all the blocks are replicated in other nodes as well for fault tolerance.

- A file of size 513 MB will be stored as 5 blocks, the first four blocks of 128 MB each and the fifth block will be of size 1 MB.

- **Block size can be changed using the below properties in hdfs-site.xml.**

```
<property>
<name>dfs.block.size</name>
<value>134217728</value>
<description>Block size</description>
</property>
```

HDFS High Level Architecture



Processing Constraint with HDFS block?

Let's take a scenario:

- We have a large file and each line in the file represents one record.

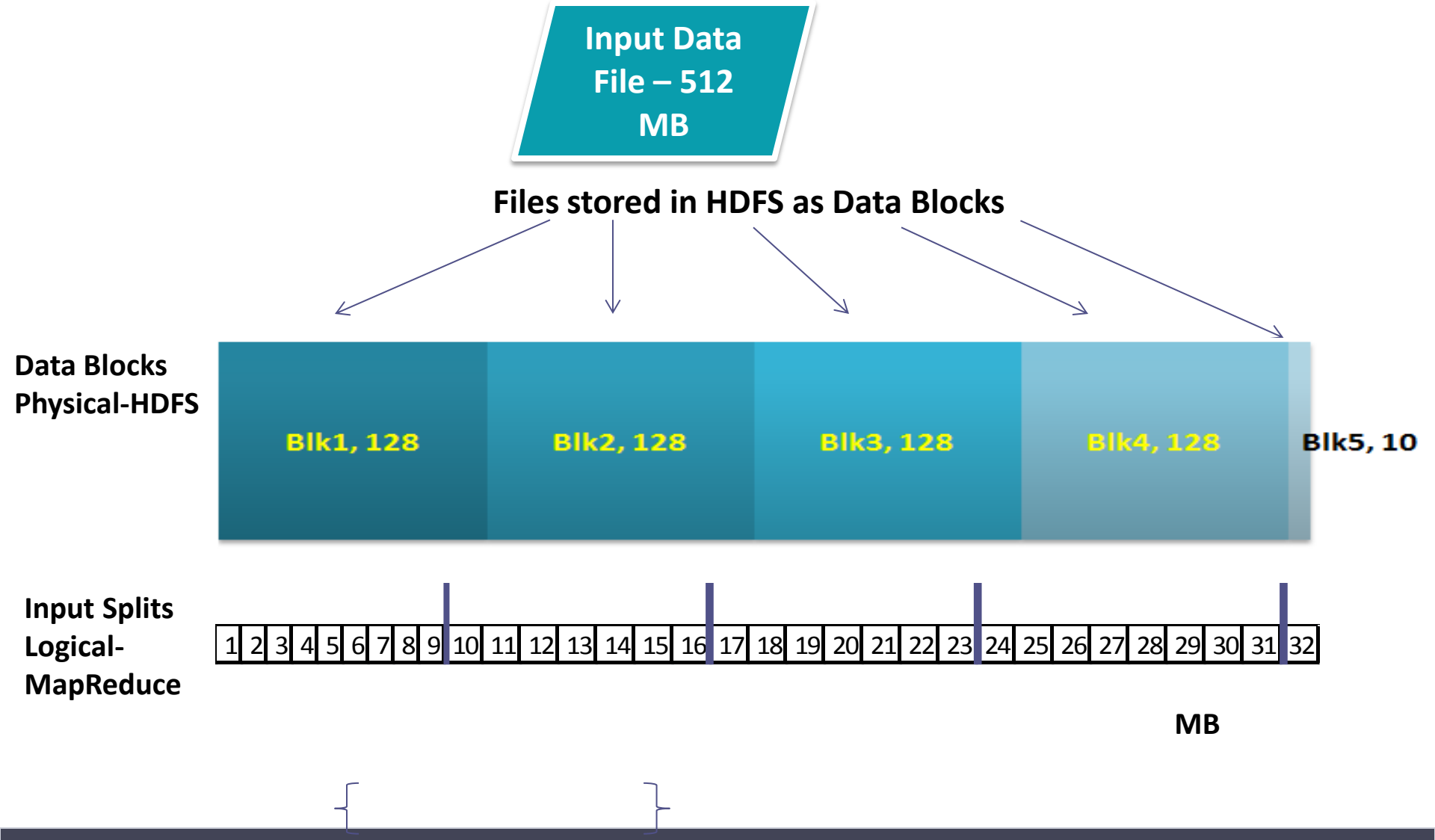
- If each Map task processes all records in a specific data block then how will the records spanning across block boundaries be processed?

Here is the Solution!

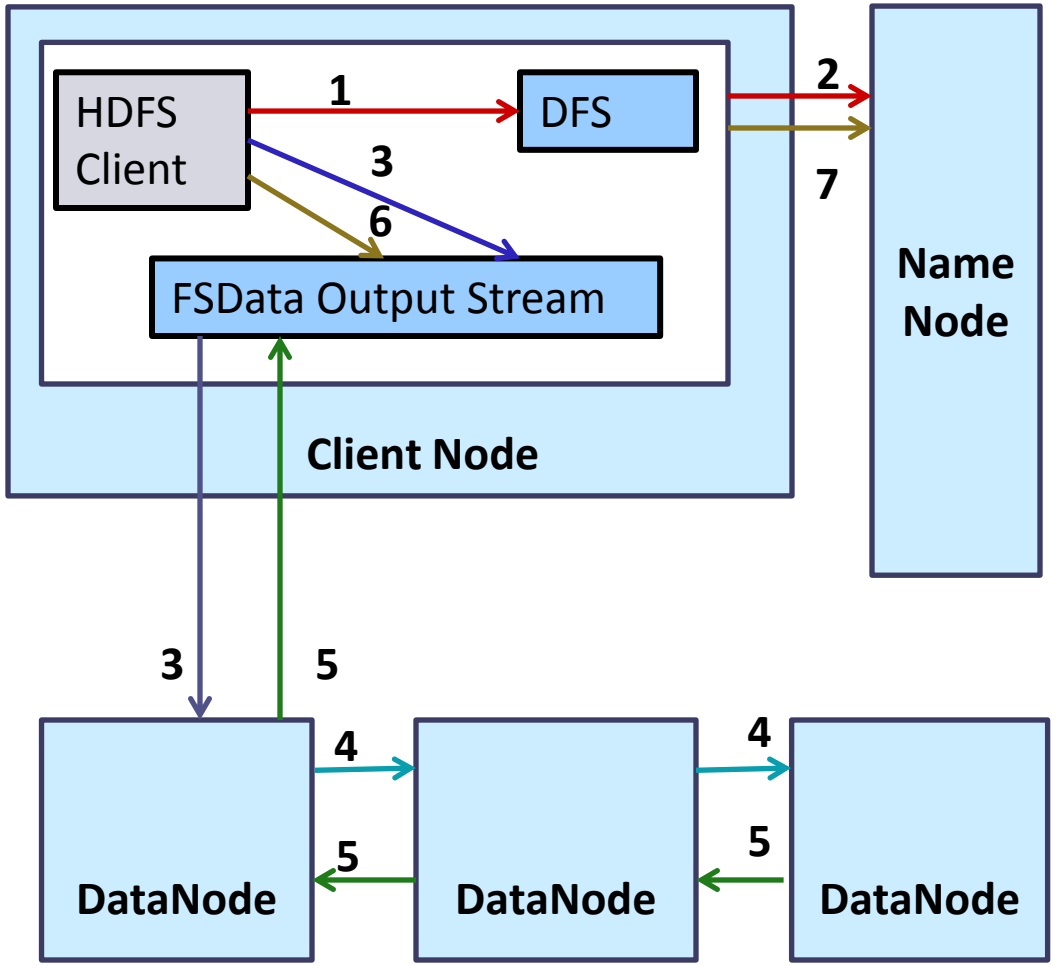
Solution: Processing on Input split

- ✓ Hadoop uses the logical representation of the data stored in the file blocks known as Input splits.
- ✓ MapReduce Job client calculates the input splits by figuring out the location of first whole record in the block and the location where the last record in the block ends.
- ✓ In cases where the last record in a block is incomplete, the input split includes location information of the next block and the byte information of the data needed to complete the record.
- ✓ MapReduce programs are written in higher layers where details of underlying storage are abstracted out – i.e details are hidden to reduce complexity.

Relationship between HDFS Blocks and Input Splits



HDFS – File Write



1. Create file by calling **Create()** method on DFS.
2. DFS calls NameNode to create file in File System's Namespace.
3. Data packets are streamed to first DataNode in the pipeline.
4. Data is streamed to 2nd & 3rd DataNode.
5. Success/Failure Ack.
6. When the client has finished writing data, it calls **close()** on the stream.
7. After receiving Acks for copied packets to NameNode - Contacting the NameNode to signal that the file is complete.

Questions - HDFS

- What is HDFS?
- What are the main operations of HDFS?
- How does data get distributed by HDFS write command?
- What is limitation with processing on HDFS block.
- Input split is logical or physical division?

Next Session (4)

- **Data Model:**
  - File Read
- **Hadoop Installation:**
  - Hadoop configuration files
- **Demo of HDFS Commands**
- **Hadoop Eco System:**
  - Key Components

Get In touch with AcadGild

Contact Info:

- **Website : http://www.acadgild.com**
- **LinkedIn : https://www.linkedin.com/company/acadgild**
- **Facebook : https://www.facebook.com/acadgild**
- **Support : support@acadgild.com**