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 Information Technology

19IT601– Data Science and Analytics

III Year / VI Semester

Unit 4 – DATA ANALYTICAL FRAMEWORKS

Topic 3: HDFS Block Replication

Block Replication

Features of Hadoop

- It can handle massive quantities of structured, semi-structured and unstructured data.
- Hadoop has shared nothing architecture
- Data Replication - It replicated data across multiple computers so if one goes down, the data can still be processed from another machine that stores replica.
- Hadoop is for high throughput rather than low latency.
- It is not good when work cannot be parallelized.
- It is not good for processing small files.

Hadoop

Block and Replication

•Generally the user data is stored in the files of HDFS. The file in a file system will be divided 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 data that HDFS can read or write is called a Block.

•The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

•HDFS provides performance through distribution of data and fault tolerance through 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.

Hadoop- Versions

Versions of Hadoop

There are two versions of hadoop available

- Hadoop 1.0
- Hadoop 2.0

Hadoop 1.0

Data Storage Framework – HDFS

Data Processing Framework – MapReduce – Cluster Resource Management and Data Processing

Hadoop 2.0

Data Storage Framework – HDFS

Data Processing Framework – MapReduce - Data Processing

Yarn – Cluster Resource Manager

HDFS – File Read

File Read

1. The client opens the file that it wishes to read from by calling open() on distributed filesystem.

2. Distributed filesystem communicates with name node to get the location of data blocks. Distributed Filesystem returns an DFSDataInputStream to client to read from the file

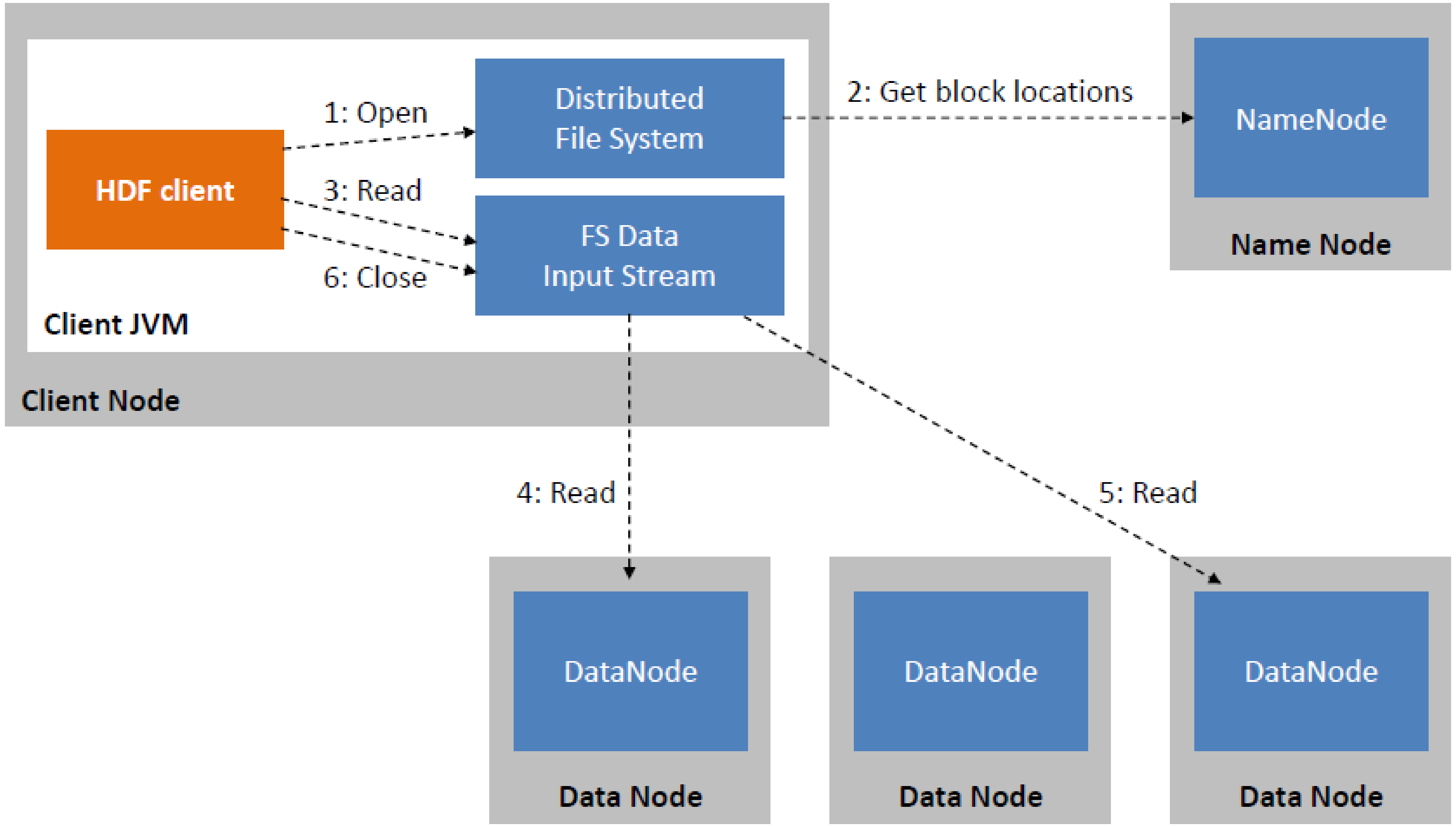
3. Client then calls read() on the stream DFSDataInputStream which has address of Data Nodes for the first few blocks of file and connects to closets Data node.

4. Client calls read() repeatedly to stream data from the Data Node.

5. When end of block is reached DFSDataInputStream closes the connection with the data node. It repeats the steps to find the best DataNode for the next block and subsequent blocks.

6. When client completes the reading of file, it calls close() on FSDataInputStream to close the connection.

HDFS – File Read

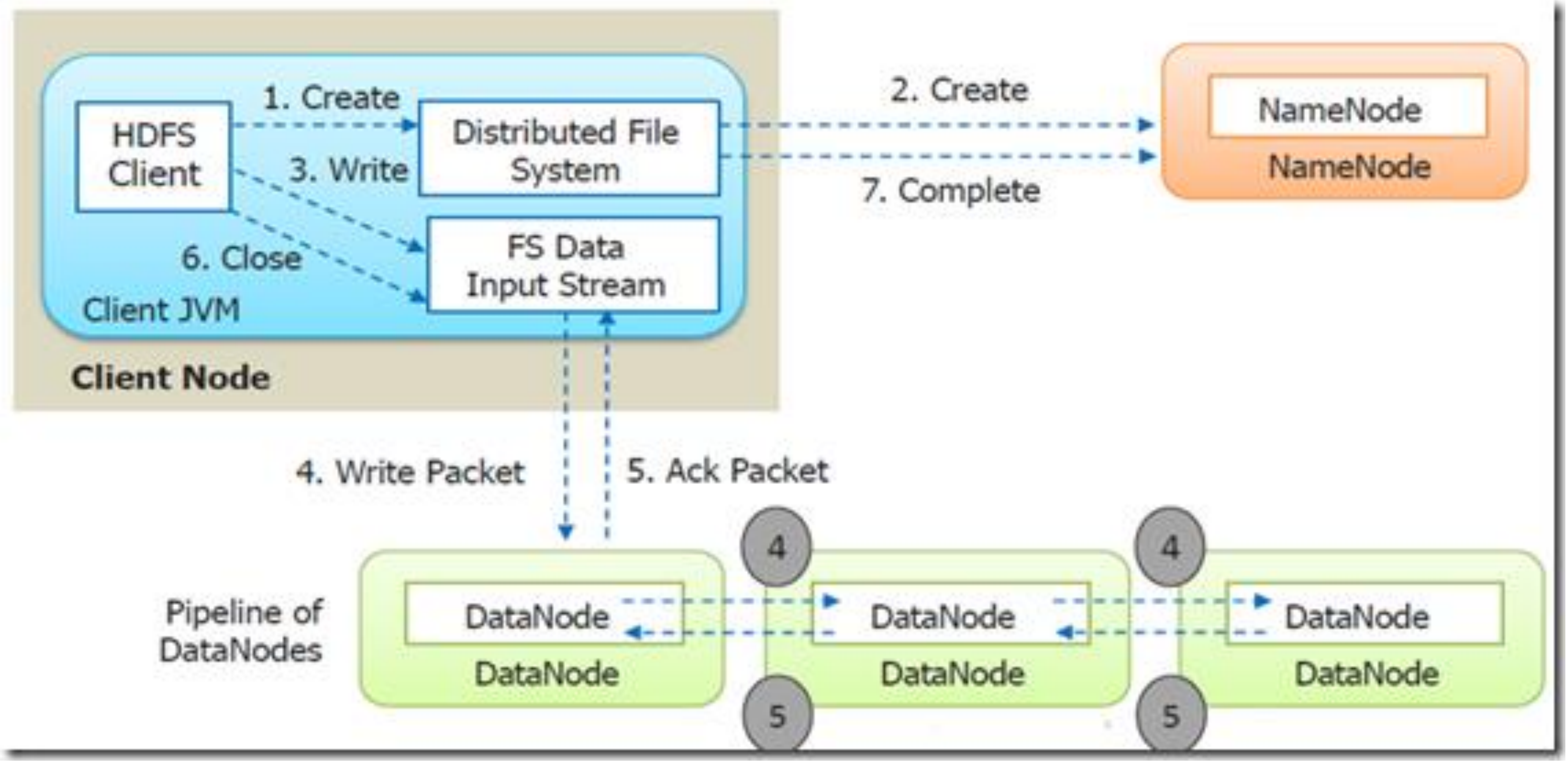


HDFS – File Write

File Write

1. The client creates the file by calling create() on DistributedFileSystem(DFS).
2. DFS makes an RPC call to the name node to create a new file in the file system's namespace, with no blocks associated with it. The DFS returns an FSDataOutputStream for the client to start out writing data to.
3. Because the client writes data, the DFSOutputStream splits it into packets, which it writes to an queue called the data queue. The data queue is consumed by the DataStreamer, which request Namenode to allocate new blocks by selecting suitable data nodes to store replicas. Three replicas require three nodes in the pipeline for the first block.
4. Datastreamer streams the packet to the first data node in the pipeline. It stores packet and forwards it to second data node in the pipeline. Further Second data node stores data and forwards it to the third data node in the pipeline.
5. The DFSOutputStream sustains an internal queue of packets that are waiting to be acknowledged by data nodes, called an "ack queue".
6. When the client finishes writing the file it calls close() on the stream
7. This flushes the remaining packets to the data node pipeline and waits for relevant ack before communicating with name node to inform the client of the file is complete.

HDFS – File Write



Hadoop vs SQL

Hadoop	SQL
Scale out	Scale up
Key value pairs	Relational Table
Functional Programming	Declarative Queries
Off-line batch processing	Online transaction processing

THANK YOU