HDFS

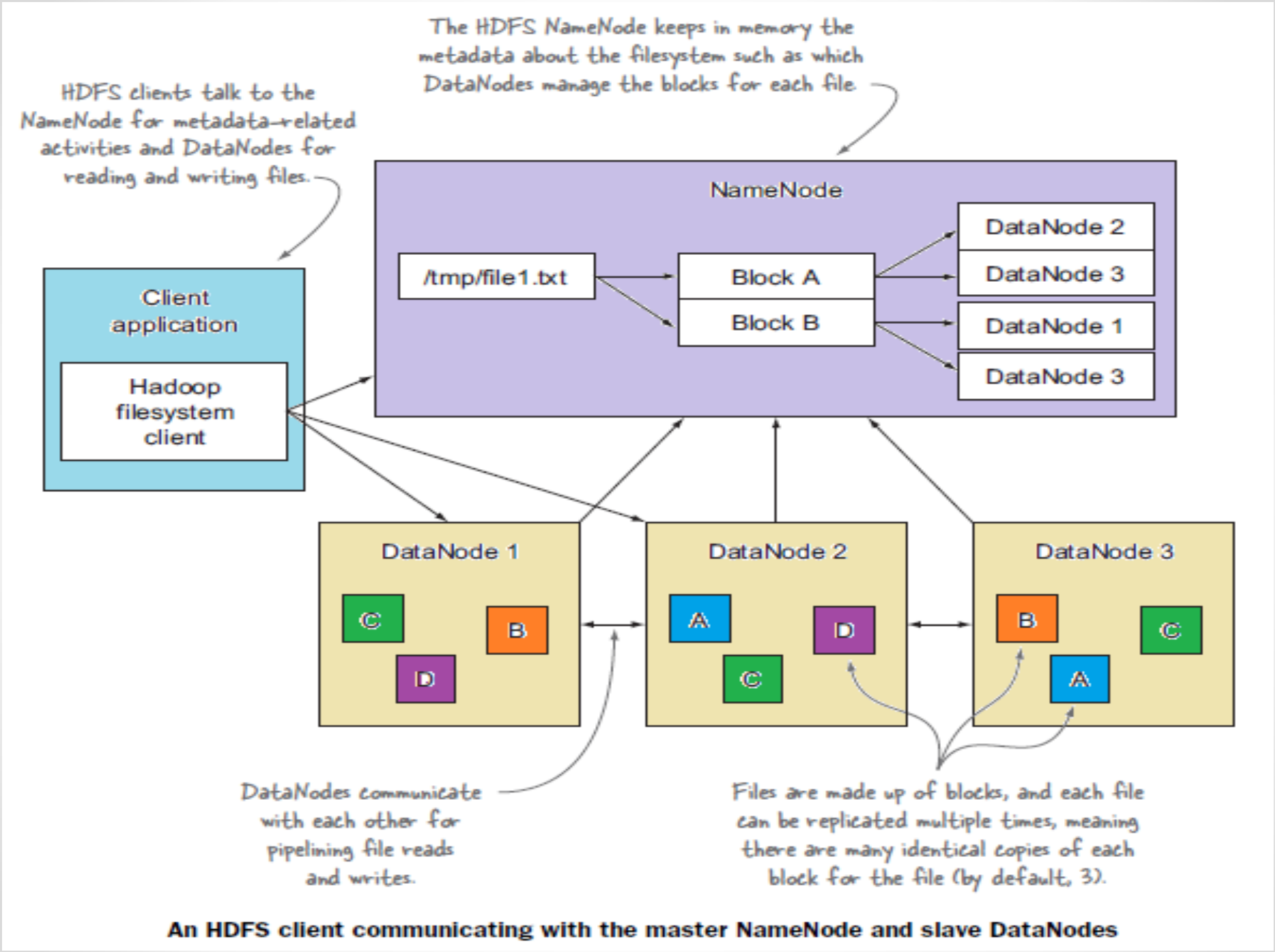
Hadoop Core Components

- **HDFS:**
  (a) Storage component.
  (b) Distributes data across several nodes.
  (c) Natively redundant.

- **MapReduce:**
  (a) Computational framework.
  (b) Splits a task across multiple nodes.
  (c) Processes data in parallel.

Hadoop Distributed File System

- Storage component of Hadoop.
- Distributed File System.
- Modeled after Google File System.
- HDFS stores files in blocks typically at least 64 MB in size, much larger than the 4-32 KB seen in most filesystems.
- Optimized for throughput over latency. Very efficient at streaming read requests for large files but poor at seek requests for many small ones.
- Data locality - Move computation close to where data is stored
- You can replicate a file for a configured number of times, which is tolerant in terms of both software and hardware.
- Re-replicates data blocks automatically on nodes that have failed.



An HDFS client communicating with the master NameNode and slave DataNodes

NameNode

- Single NameNode per cluster.
- Manages the filesystem namespace.
- Maintains the filesystem tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files:
  - **FsImage** – contains entire file system namespace, including the mapping of blocks to files and file system properties
  - **EditLog** – contains every change that occurs to file system metadata
- Does not store block locations persistently, because this information is reconstructed from datanodes when the system starts.

Secondary Namenode

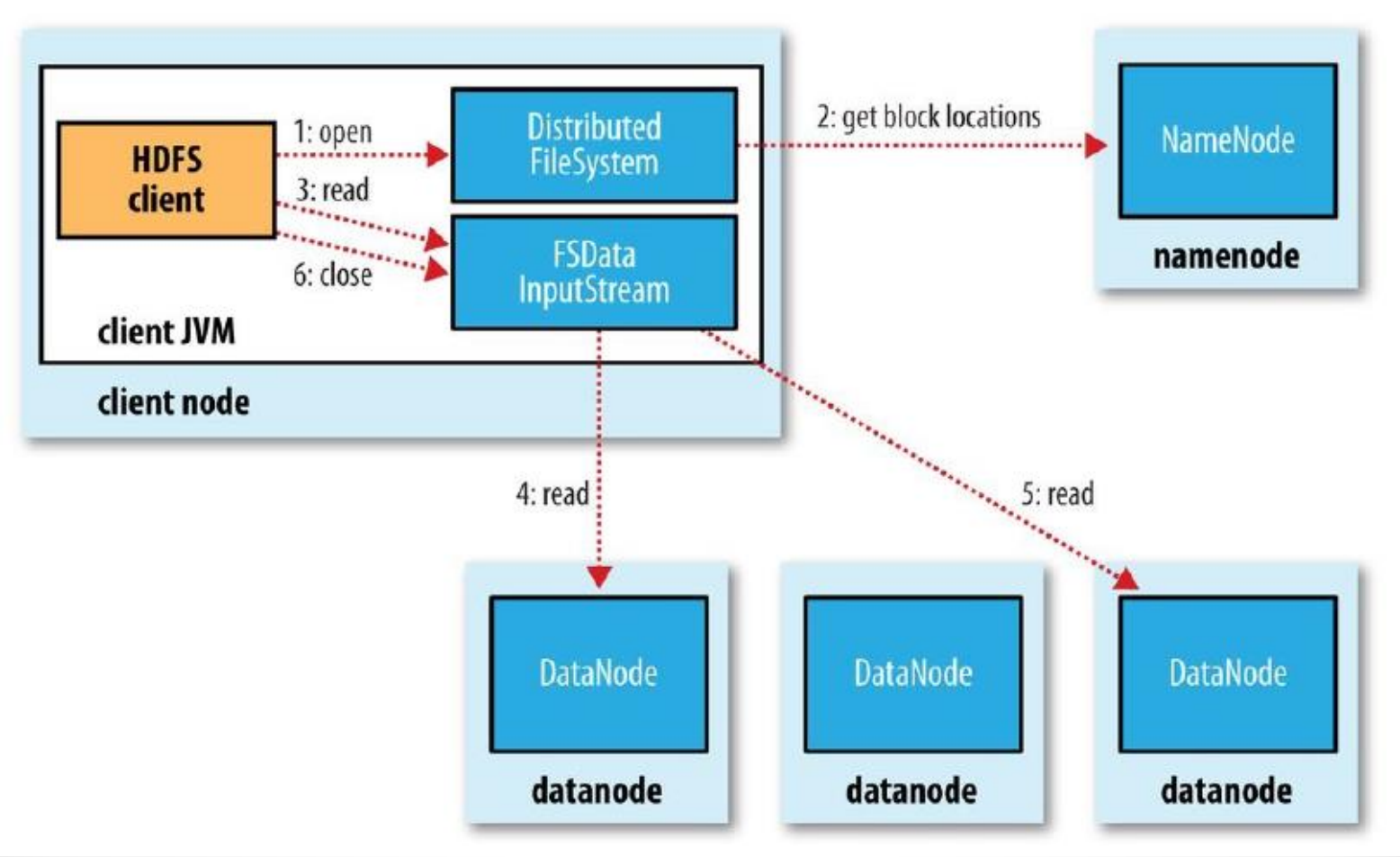
(a poorly named component of hadoop)

- Not backup node but checkpoint node
- Periodically merges the FsImage with the edit log to prevent the edit log from becoming too large. Required for NameNode to start faster next time.
- Keeps a copy of the merged FsImage, which can be used in the event of the namenode failing.
- It is possible to run a hot standby namenode instead of a secondary for High Availability in case of version 2.6 or higher

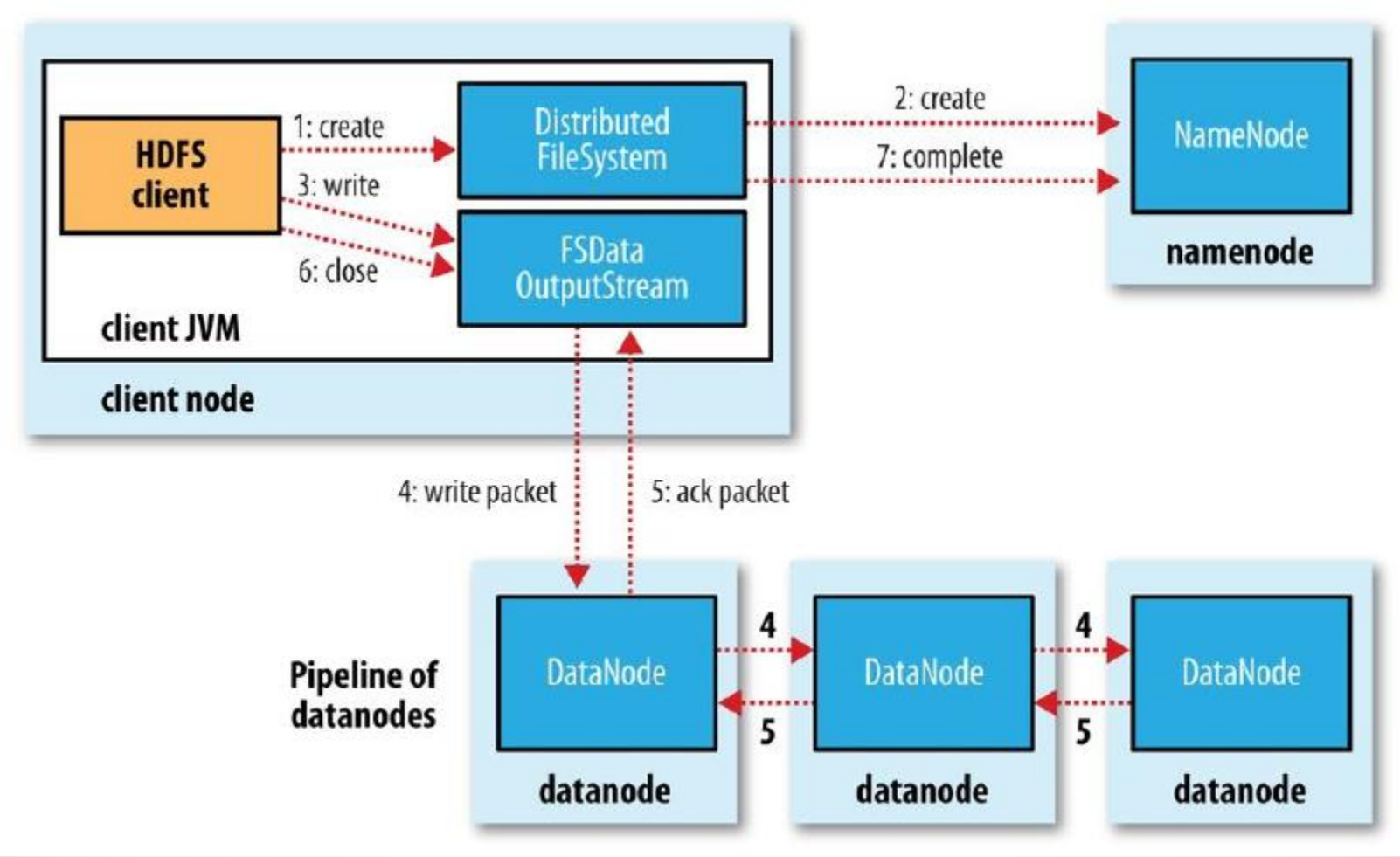
DataNode

- Multiple DataNode per cluster
- Stores HDFS data in files in its local file system
- Has no knowledge about HDFS files
- On start up, scans through its local file system, generates a list of all HDFS data blocks and sends this report to the NameNode
- Communicate with each other during pipelined read/write
- Sends periodic heartbeat message to "NameNode"
  - Default is heartbeat every 3 seconds and block report every hours

Anatomy of File Read

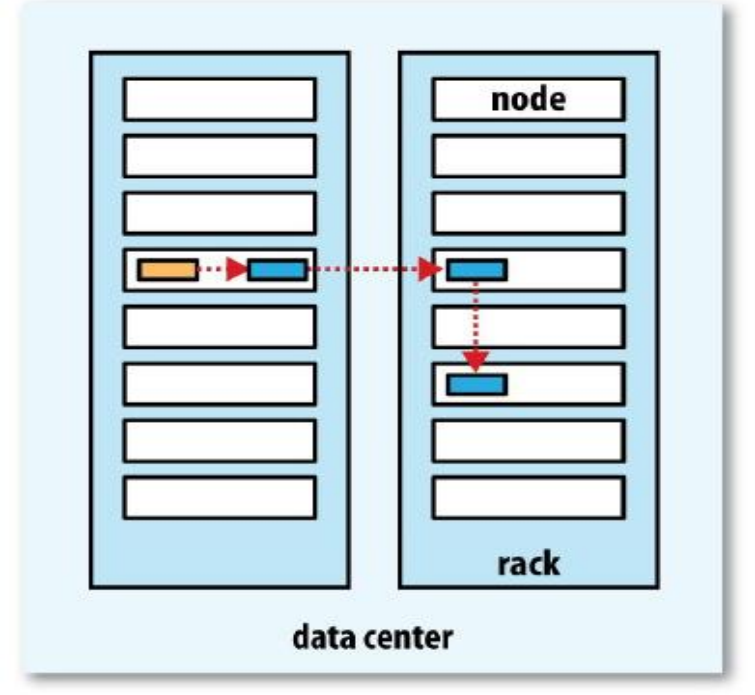


Anatomy of File Write



Replica Placement Strategy

- First Replica – Same Node as the client(for clients running outside the cluster, a node is chosen at random)
- Second Replica – Node on different rack
- Third Replica - same rack as second, but on a different node in the rack



Special Features of HDFS

- **Data Replication:** There is absolutely no need for a client application to track all blocks. It directs the client to the nearest replica to ensure high performance.

- **Data Pipeline:** A client application writes a block to the first DataNode in the pipeline. Then this DataNode takes over and forwards the data to the next node in the pipeline. This process continues for all the data blocks, and subsequently all the replicas are written to the disk.

HDFS Commands

- **Objective:** To create a directory (say, sample) in HDFS.
-
- **Act:**
-
- ***hadoop fs -mkdir /sample***
- **Objective:** To copy a file from local file system to HDFS.
- **Act:**
- ***hadoop fs -put /root/sample/test.txt /sample/test.txt***
- **Objective:** To copy a file from HDFS to local file system.
- **Act:**
- ***hadoop fs -get /sample/test.txt /root/sample/testsample.txt***