

HCIA Big Data HDFS – Hadoop Distributed File System & ZooKeeper Eng. Hagar Muhammed

Lecture 2



content

- HDFS overview.
- Use Cases of HDFS
- HDFS Architecture
- HDFS Key Features
- HDFS shell Commands
- ZooKeeper Overview
- ZooKeeper Architecture
- Summary.

Contents

1. HDFS

- HDFS Overview
 - HDFS characteristics
 - HDFS Architecture
 - HDFS Key Features
 - HDFS Data Read/Write Process

2. ZooKeeper Distributed Coordination Service



Hadoop distributed file system (HDFS) is developed based on Google file system (GFS) and runs on commodity hardware.

- HDFS is part of the Apache Hadoop Core project.
- IT creates multiple replica of data blocks and distributed them on nodes in a cluster.
- Distribution enables rapid computations.

Contents

1. HDFS

- HDFS Overview
- HDFS-characteristics
- HDFS Architecture
- HDFS Key Features
- HDFS Data Read/Write Process

2. ZooKeeper Distributed Coordination Service

HDFS characteristics

Distributed

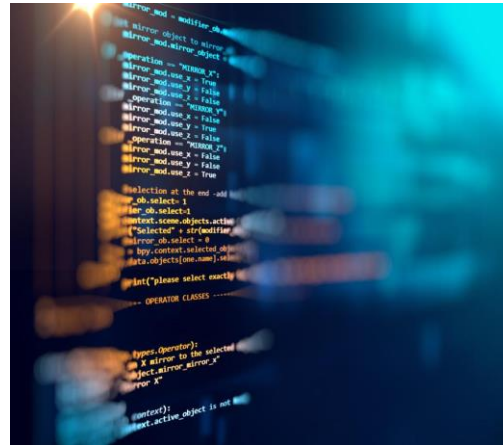
High Throughput

Fault Tolerant

scalable

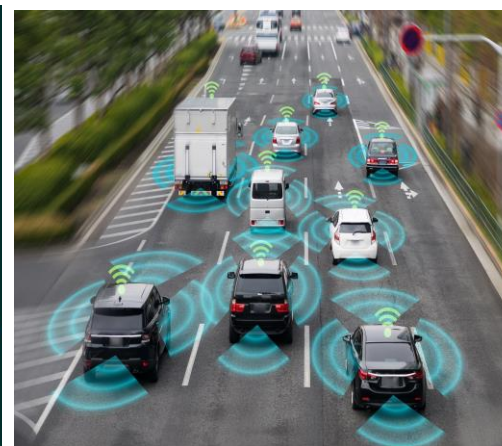
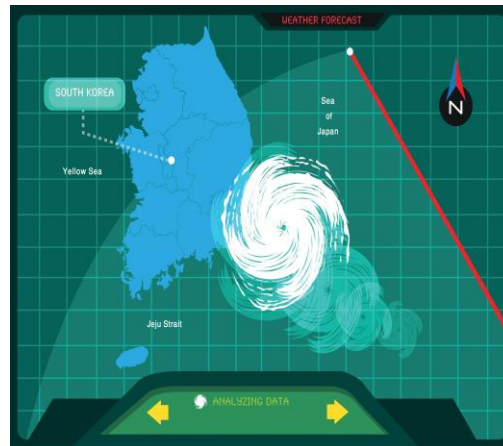
Cost effective

Website data storage



Traffic Monitoring data storage

Meteorological data storage



Sensor data storage

Contents

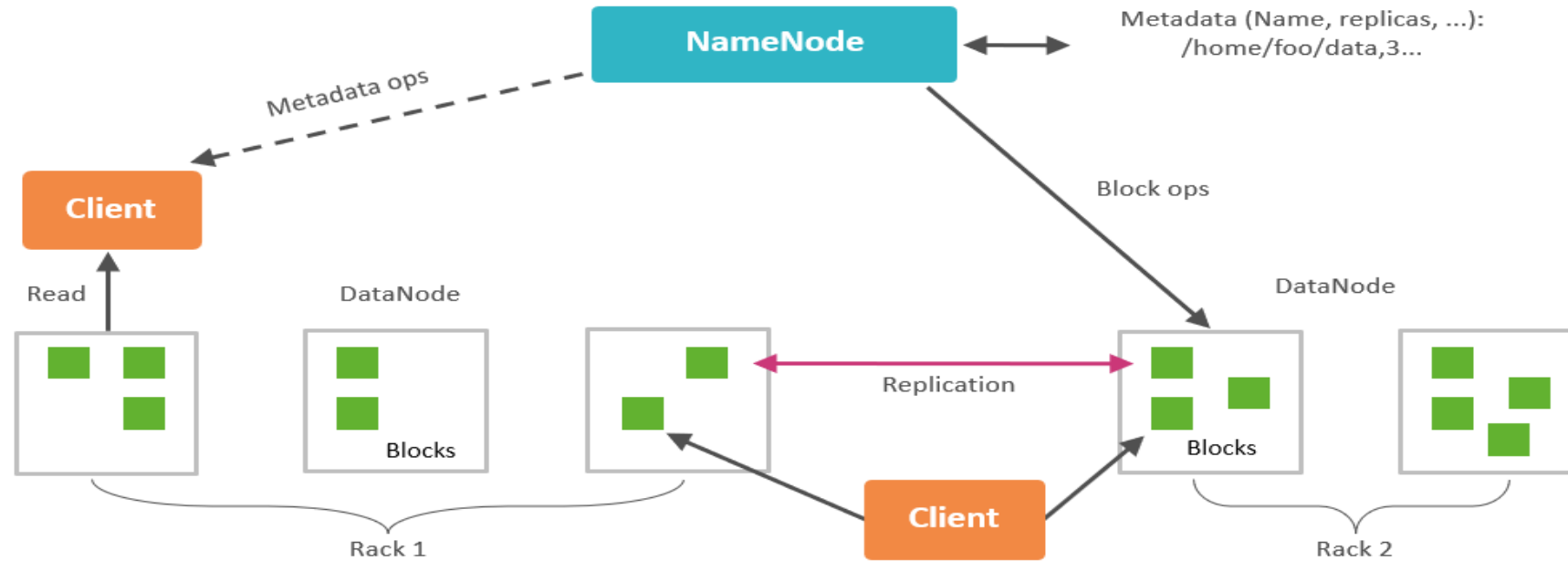
1. HDFS

- HDFS Overview
- HDFS-characteristics
- HDFS Architecture
- HDFS Key Features
- HDFS Data Read/Write Process

2. ZooKeeper Distributed Coordination Service



Basic HDFS System Architecture



Client

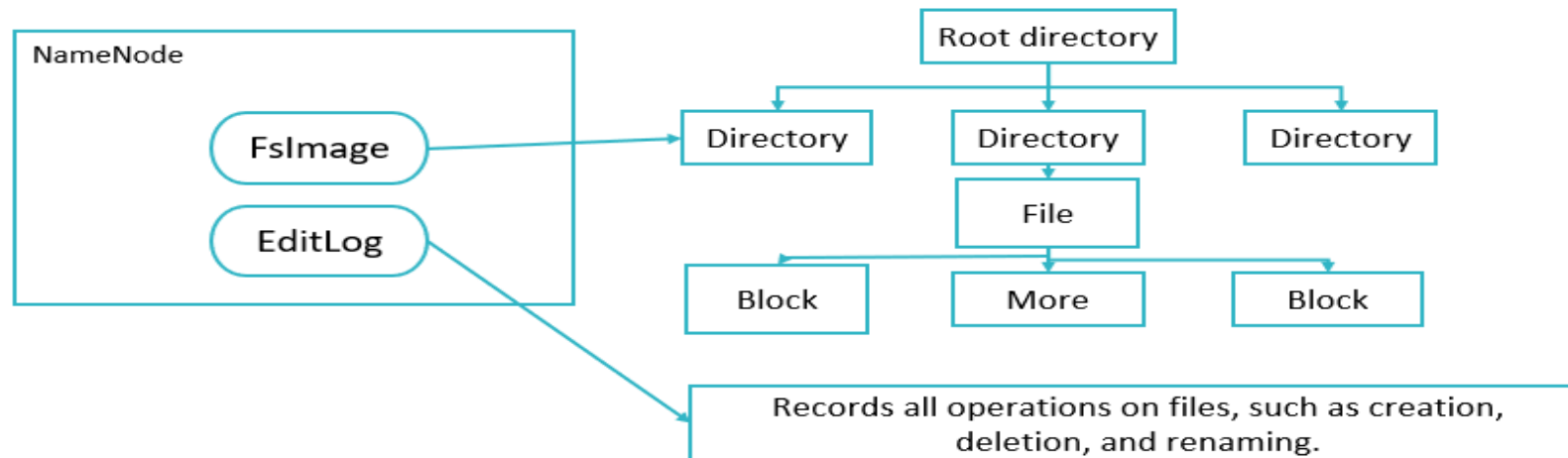
- It is a library that contains HDFS interfaces that hide most of the complexity in HDFS implementation
- It supports common operations such as opening, reading, and writing, and provides a shell-like command line mode to access data in HDFS.

NameNode

In HDFS, the NameNode manages the namespace of the distributed file system and stores two core data structures: FsImage and EditLog.

FsImage maintains the metadata of the file system tree and all the files and folders in that file tree.

EditLog records all the operations on files, such as creation, deletion, and renaming.



DataNode

- A DataNode is a worker node of HDFS. It stores and retrieves data based on the scheduling of Clients or NameNodes, and periodically sends the list of blocks stored on it to NameNodes

NameNodes vs. DataNodes

NameNode	DataNode
Stores metadata.	Stores file content.
Stores metadata in the memory.	Stores file content in the disk.

Block

- The default size of a block is 128 MB. A file is divided into multiple blocks. A block is the storage unit.
- The block size is much larger than that of a common file system
- A block has the following benefits:
 - Large-scale file storage
 - Simplified system design
 - Data backup

Disadvantages of the HDFS Single-NameNode Architecture

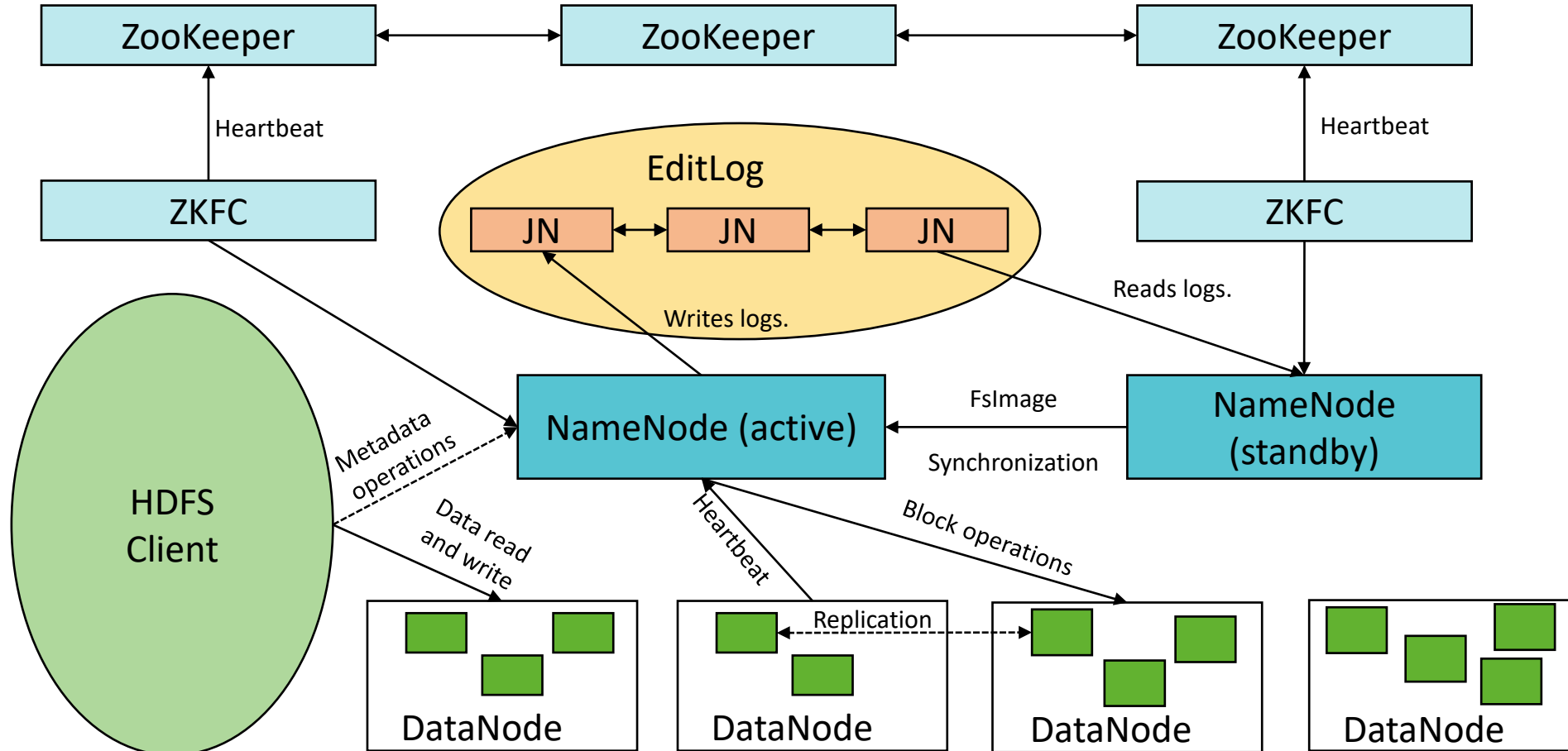
- Only one NameNode is configured for HDFS, which significantly simplifies the system's design but also has the following disadvantages:
 - **Namespace limitation:** NameNodes are stored in the memory. Therefore, the number of objects (files and blocks) that can be contained in a NameNode is limited by memory size.
 - **Performance bottleneck:** The throughput of the entire distributed file system is limited by the throughput of a single NameNode.
 - **Isolation:** Because there is only one NameNode and one namespace in the cluster, different applications cannot be isolated.
 - **Cluster availability:** Once the only NameNode is faulty, the entire cluster becomes unavailable.

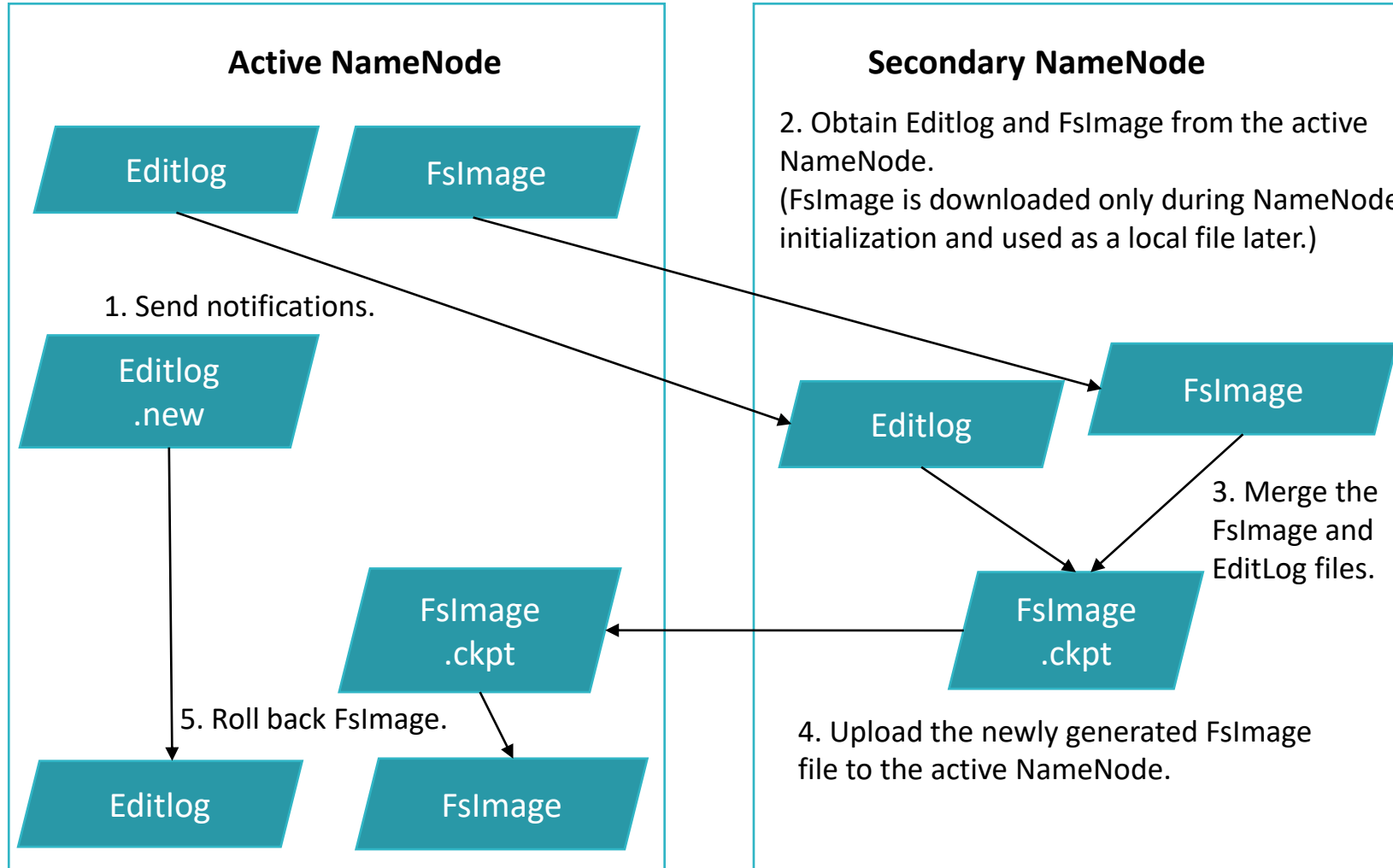
Contents

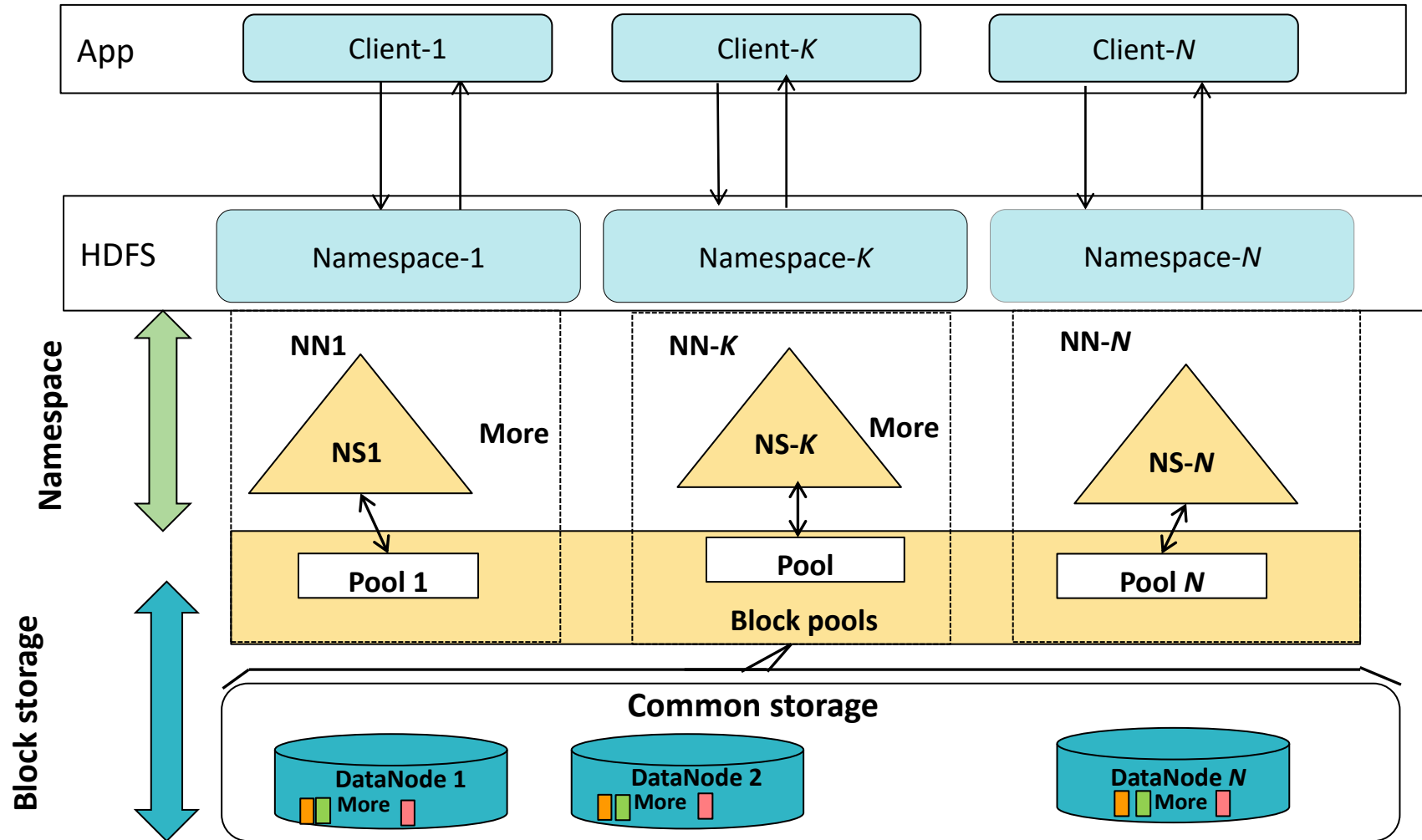
1. HDFS

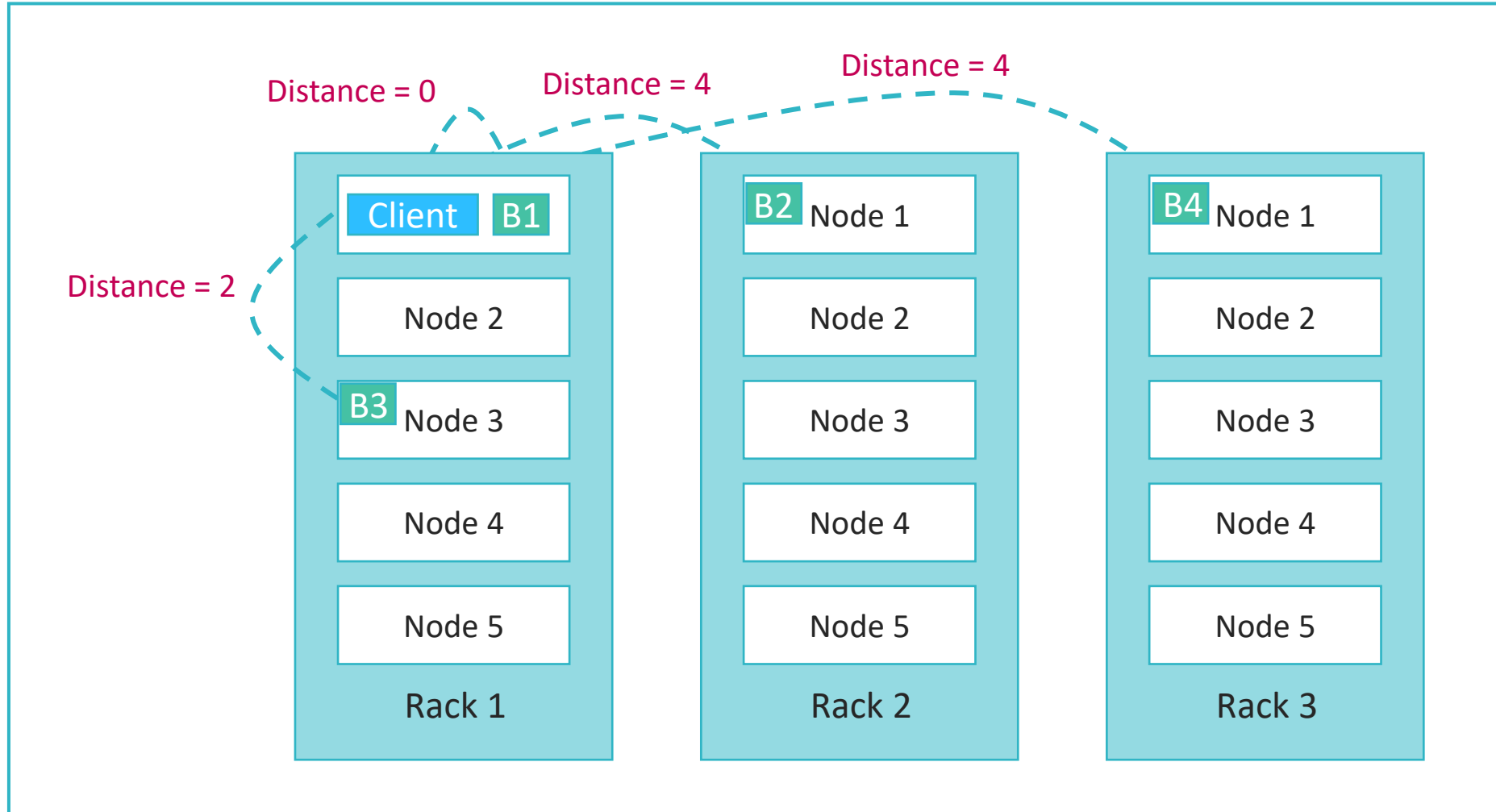
- HDFS Overview
- HDFS-Characteristics
- HDFS Architecture
- HDFS Key Features
- HDFS Data Read/Write Process

2. ZooKeeper Distributed Coordination Service









Contents

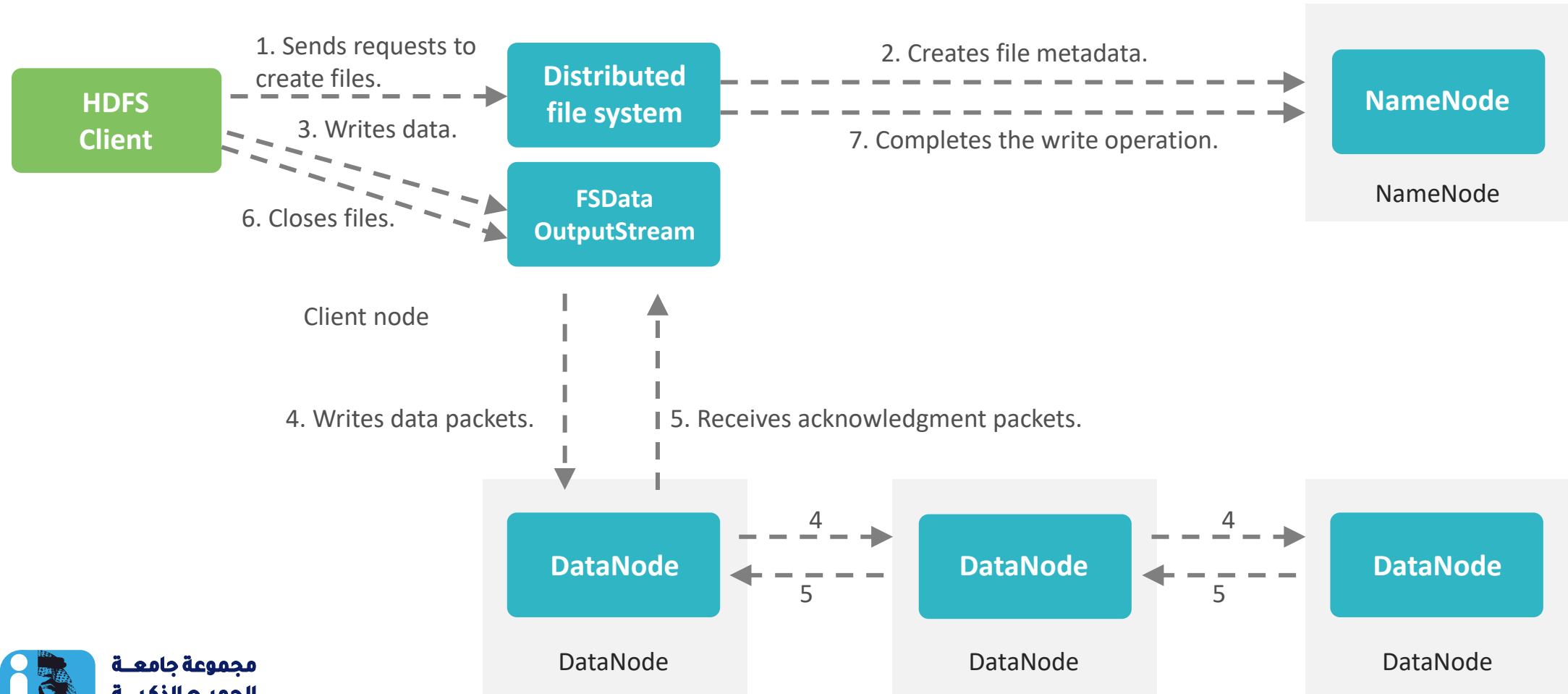
1. HDFS

- HDFS Overview
- HDFS- Characteristics
- HDFS Architecture
- HDFS Key Features
- HDFS Data Read/Write Process

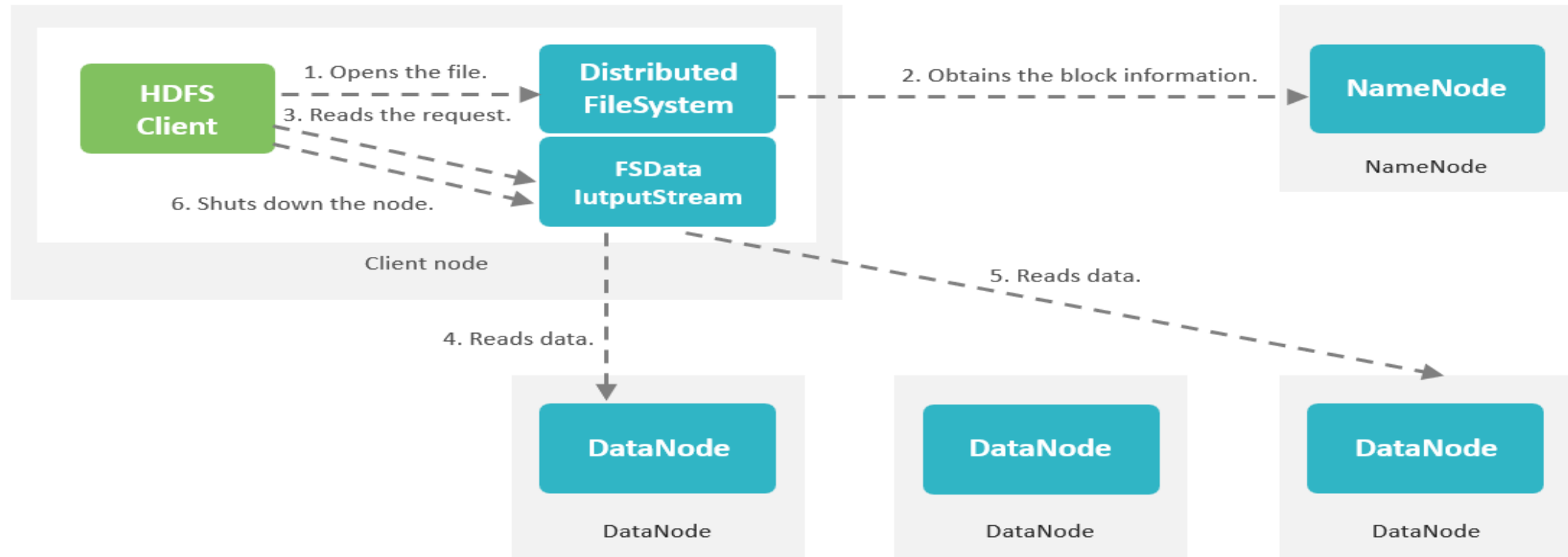
2. ZooKeeper Distributed Coordination Service



HDFS Data Write Process



HDFS Data Read Process



Common Shell Commands

Type	Command	Description
dfs	-cat	Displays file content.
	-ls	Displays the directory list.
	-rm	Deletes a file.
	-put	Uploads directories or files to HDFS.
	-get	Downloads directories or files to the local host from HDFS.
	-mkdir	Creates a directory.
	-touchz	Changes the group of a file.
	...	...
dfsadmin	-safemode	Performs operations in security mode.
	-report	Reports service status.

Contents

1. HDFS

2. ZooKeeper Distributed Coordination Service

- ZooKeeper Overview
- ZooKeeper Architecture



Zookeeper Overview

- The ZooKeeper distributed service framework is used to solve some data management problems that are frequently encountered in distributed applications and provide distributed and highly available coordination service capabilities
- As an underlying component, ZooKeeper is widely used and depended upon by upper-layer components, such as Kafka, HDFS, HBase, and Storm.
- In security mode, ZooKeeper depends on Kerberos and LdapServer for security authentication

Contents

1. HDFS

2. ZooKeeper Distributed Coordination Service

- Zookeeper Overview
- ZooKeeper Architecture



ZooKeeper Architecture

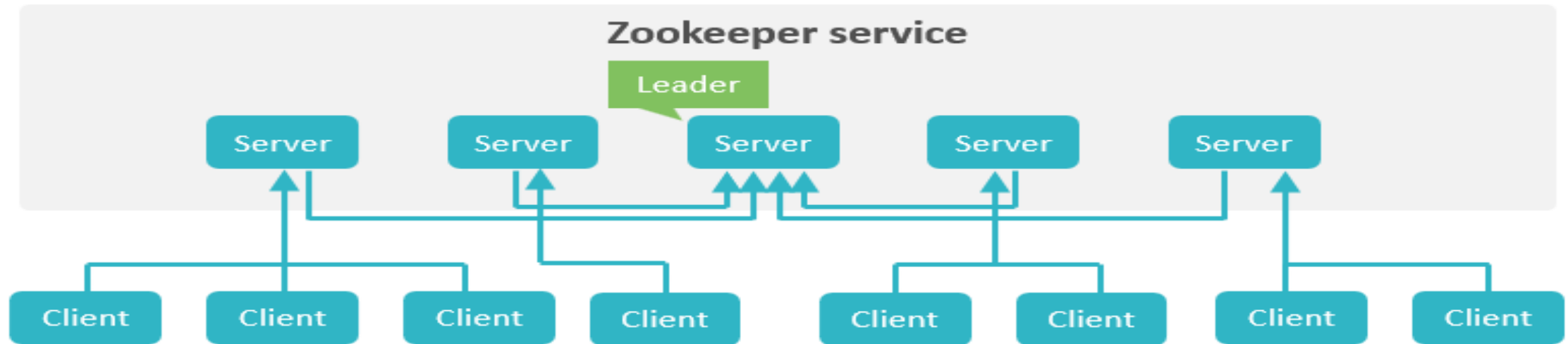
A ZooKeeper cluster consists of a group of servers. In this group, there is only one leader node, with the other nodes being followers.

The leader is elected during startup.

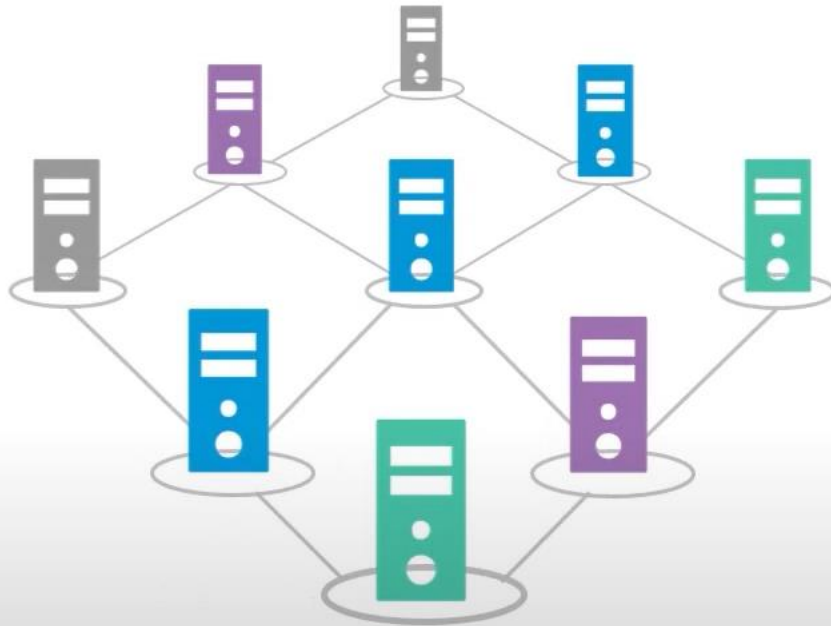
ZooKeeper uses the custom atomic message protocol to ensure data consistency among nodes in the entire system.

After receiving a data change request, the leader node writes data to the disk and then to the memory.

ZooKeeper Architecture



ZooKeeper Disaster Recovery



ZooKeeper can select a server as the leader and provide services correctly.

- An instance that wins more than half of the votes during the election becomes the leader.

For n instances, n could be odd or even.

- If $n = 2x + 1$, the node that functions as the leader must win $x + 1$ votes and the DR capability is x .
- If $n = 2x + 2$, the node that functions as the leader must win $x + 2$ votes and the DR capability is x .

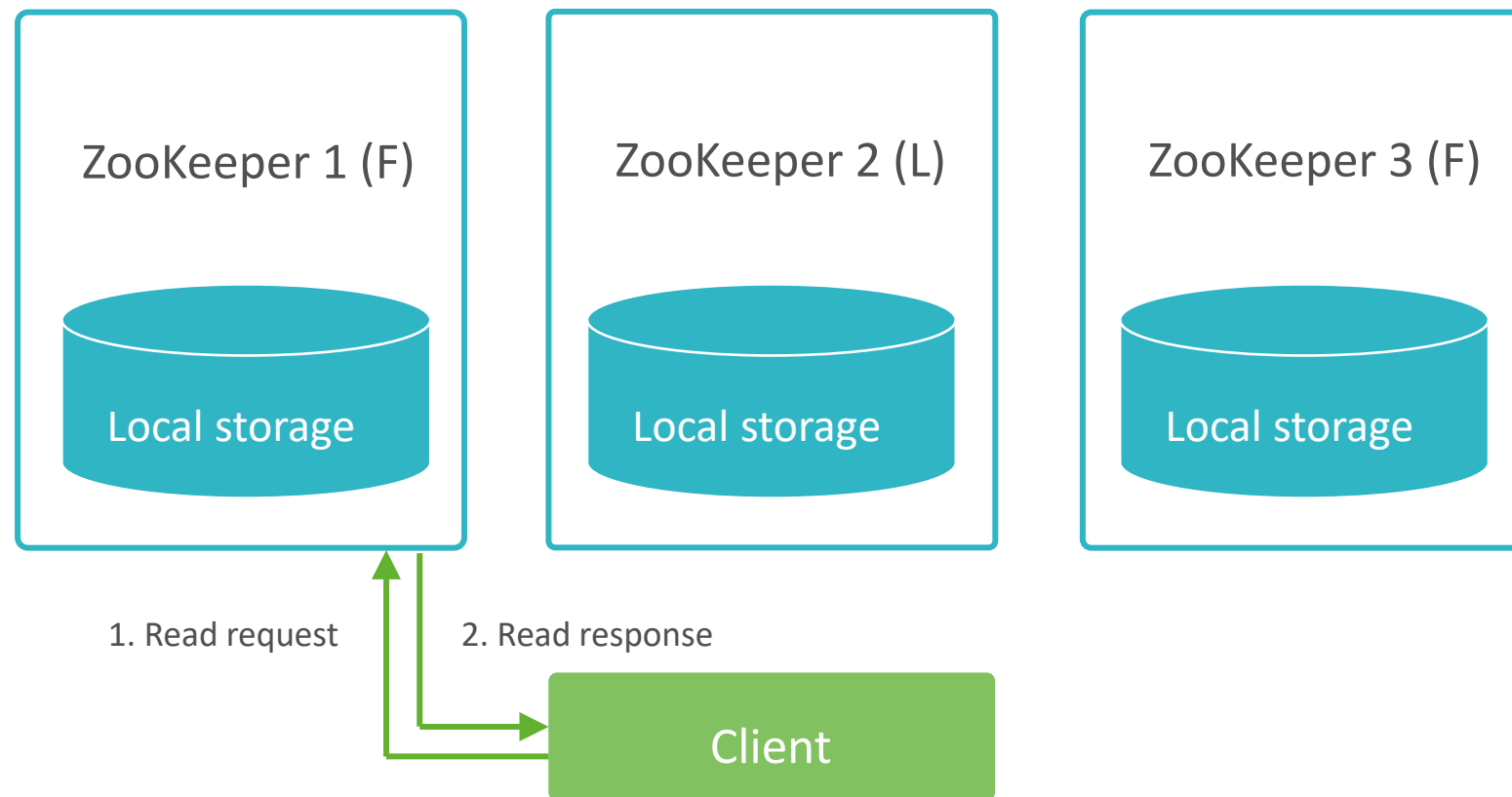
ZooKeeper key Features



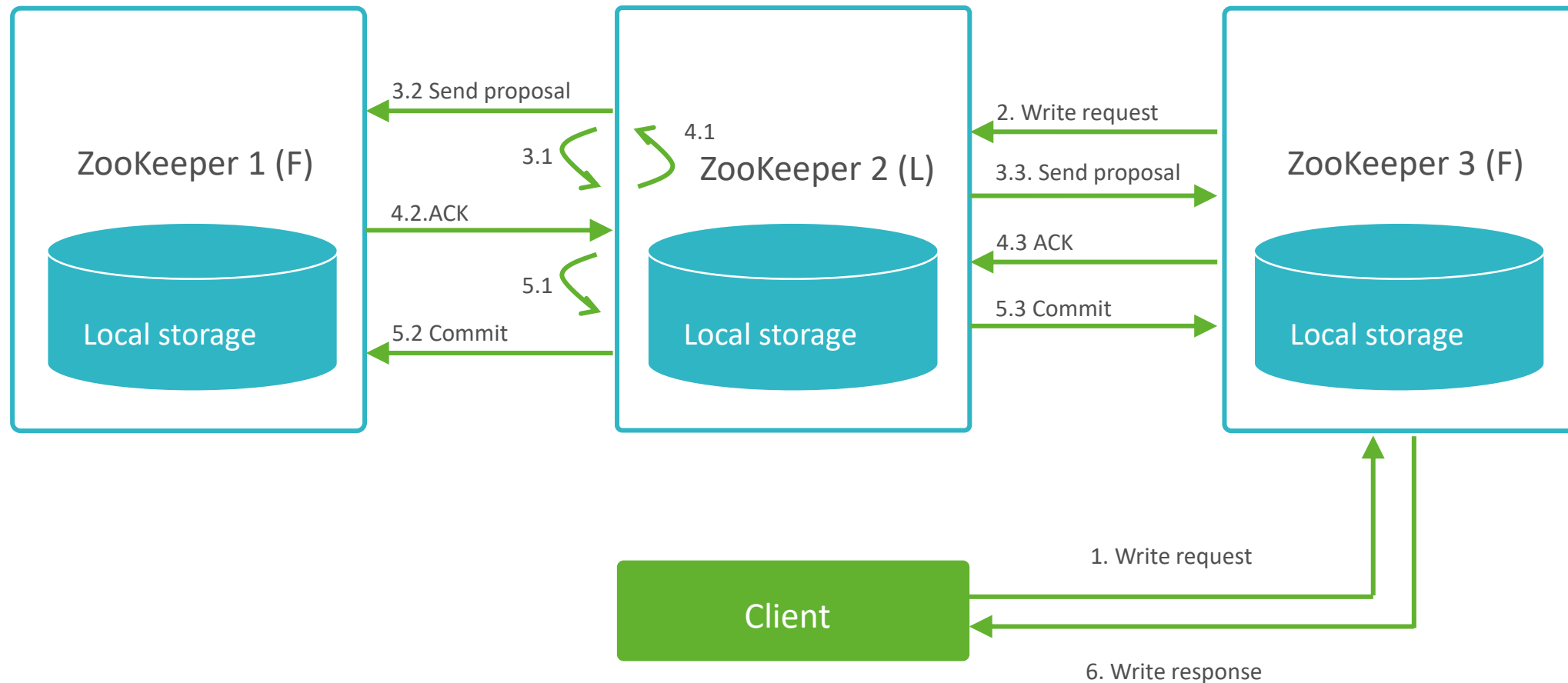
- **Eventual consistency:** All servers are displayed in the same view.
- **Real-time capability:** Clients can obtain server updates and failures within a specified period of time.
- **Reliability:** A message will be received by all servers.
- **Wait-free:** Slow or faulty clients cannot intervene the requests of rapid clients so that the requests of each client can be processed effectively.
- **Atomicity:** Data transfer either succeeds or fails, but no transaction is partial.
- **Sequence:** The sequence of data status updates on clients is consistent with that of request sending.

Read Function of ZooKeeper

- According to the consistency of ZooKeeper, the client obtains the same view regardless of the server connected to the client. Therefore, read operations can be performed between the client and any node.



Write Function of ZooKeeper



- The distributed file system is an effective solution for large-scale data storage in the era of big data. The open source HDFS implements GFS and distributed storage of massive data by using a computer cluster formed by inexpensive hardware.
- HDFS is compatible with inexpensive hardware devices, stream data read and write, large data sets, simple file models, and powerful cross-platform compatibility. However, HDFS has its own limitations.

- "Block" is the core concept of HDFS. A large file is split into multiple blocks. HDFS adopts the abstract block concept, supports large-scale file storage, simplifies system design, and is suitable for data backup.
- The ZooKeeper distributed service framework is used to solve some data management problems that are frequently encountered in distributed applications and provide distributed and highly available coordination service capabilities.

Thank You
For Your Attention