ADT302 - CONCEPTS IN BIG DATA ANALYTICS

Viju P Poonthottam
Asst. Professor
Dept. of AI & DS
MES CE Kuttippuram

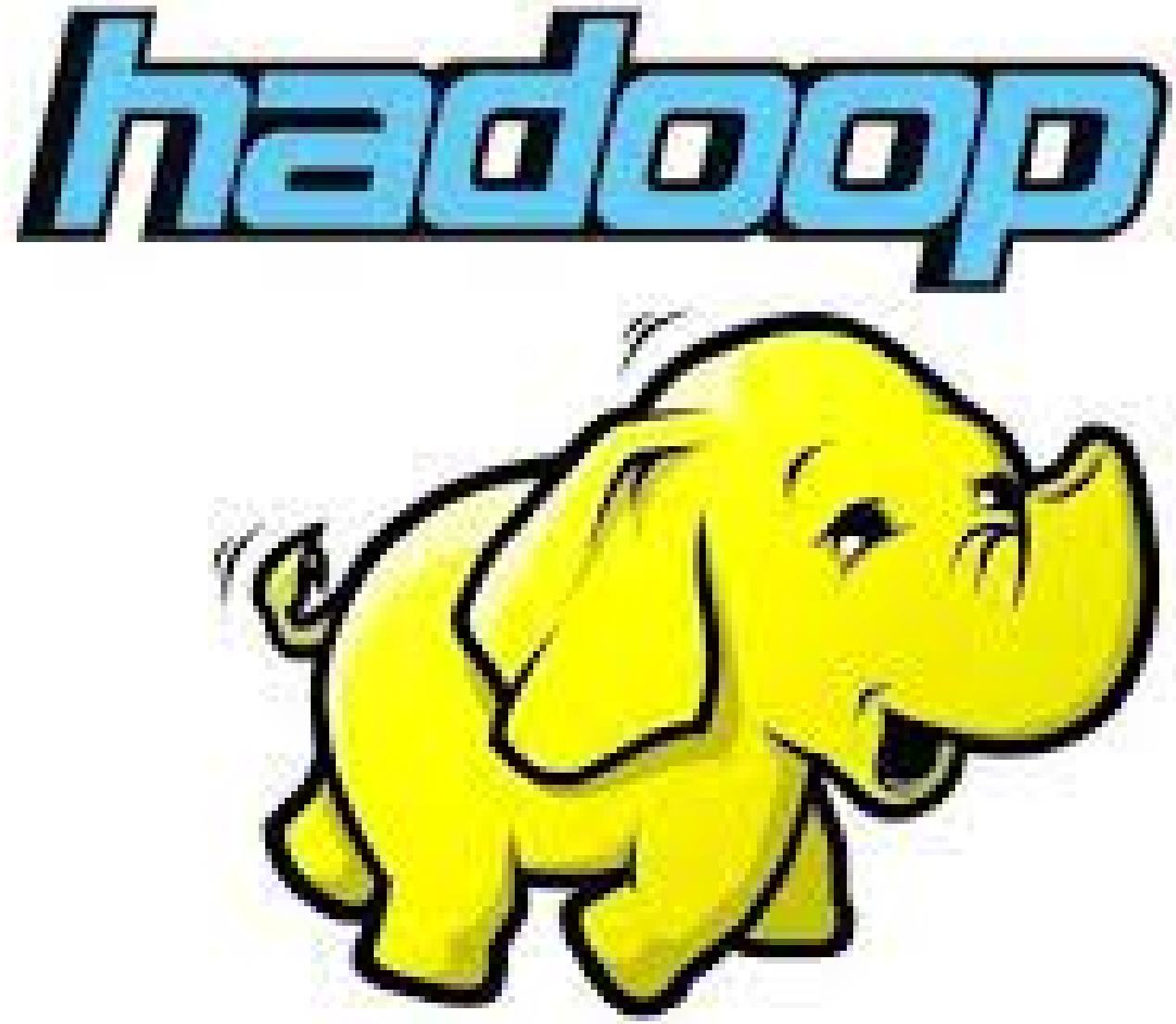
March 24, 2023

Table of contents

Module 3 -Hadoop Distributed File System(13 Hrs)

- History of Hadoop
- Hadoop Ecosystem and Core Components
- HDFS Architecture
- Using HDFS Files ,HDFS Design
- Blocks, Namenodes and Data nodes
- Basic File system Operations
- Hadoop Specific File Types
- Anatomy of a file read
- Anatomy of a file write
- Execution pipeline
- Runtime Coordination and Task Management in MapReduce
- Using MapReduce as a framework for parallel processing
- Road Enrichment Example

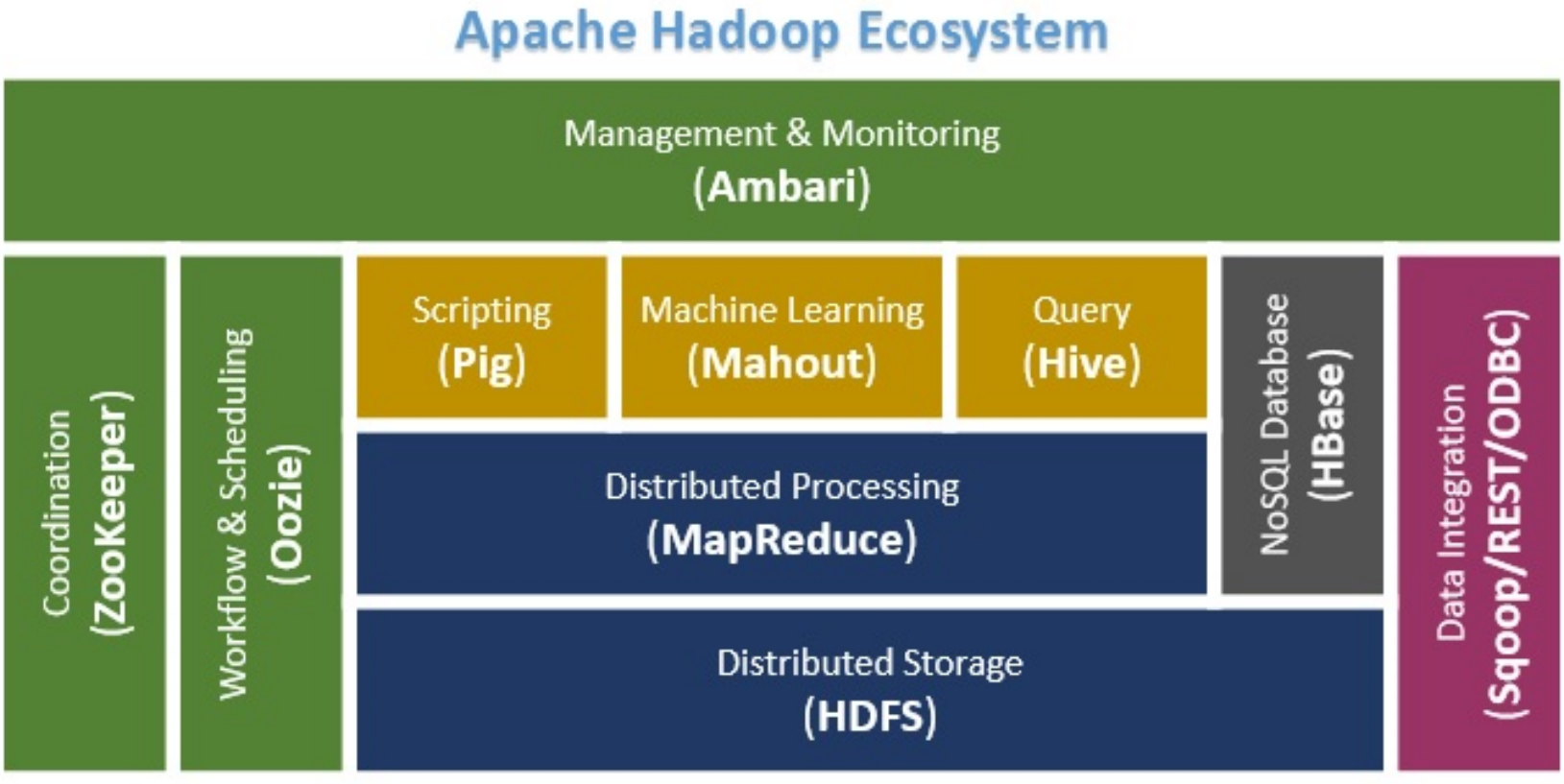
History



History

- The project's creator, Doug Cutting
- kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere
- GFS
- In 2004, Google published the paper that introduced MapReduce
- Eric Baldeschwieler- new framework, written in C++ modeled with GFS and MapReduce
- Yahoo! hired Doug Cutting,
- Hadoop was made its own top-level project at Apache
- Hadoop was being used by many other companies besides Yahoo!, such as Facebook, and the New York Times etc.

Hadoop Ecosystem and Core Components



Major Components

- HDFS - Hadoop Distributed file system
- MapReduce -Is a programming model for processing large data sets

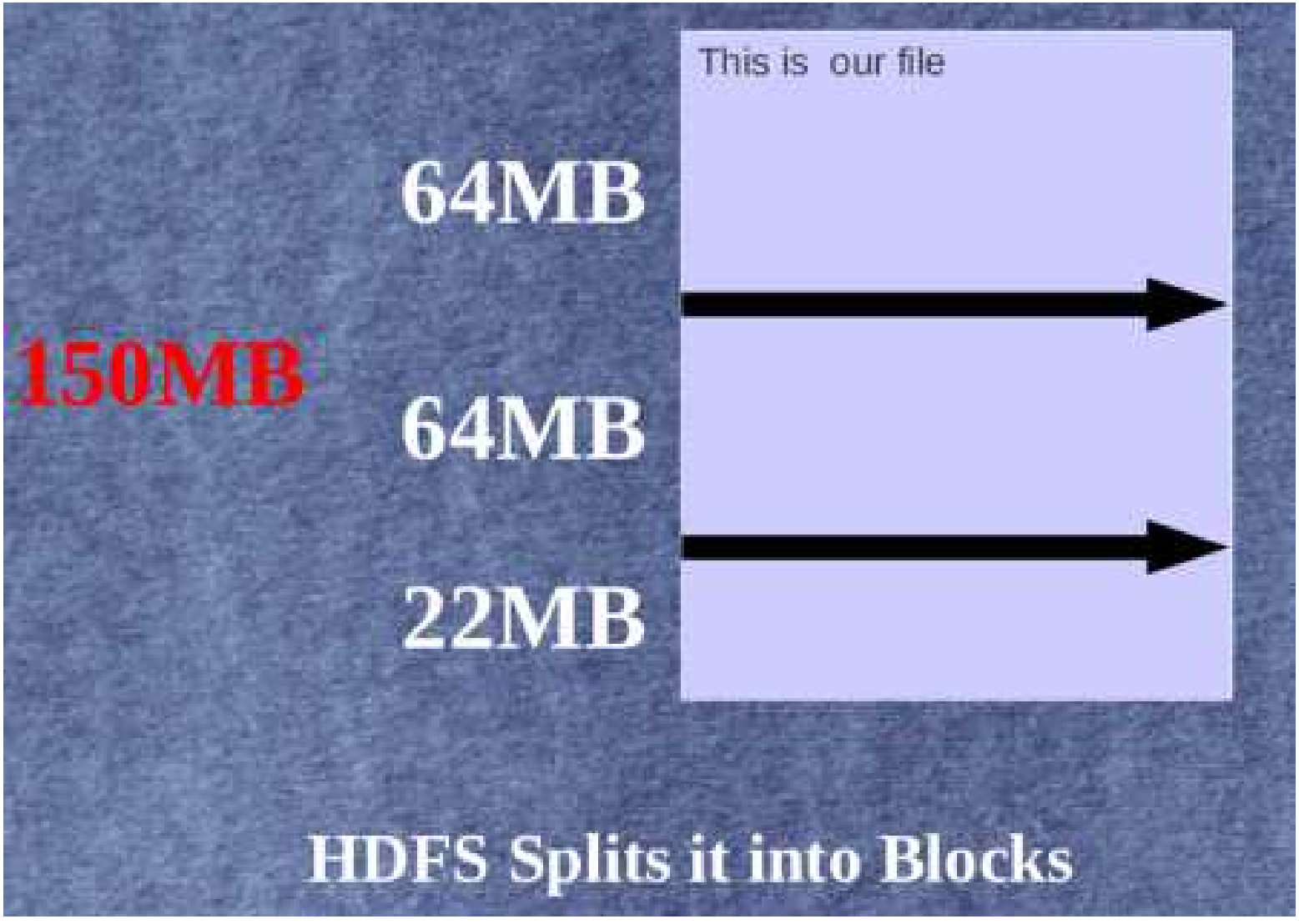
HDFS Structure

- Namenode
- Datanode
- HDFS Client

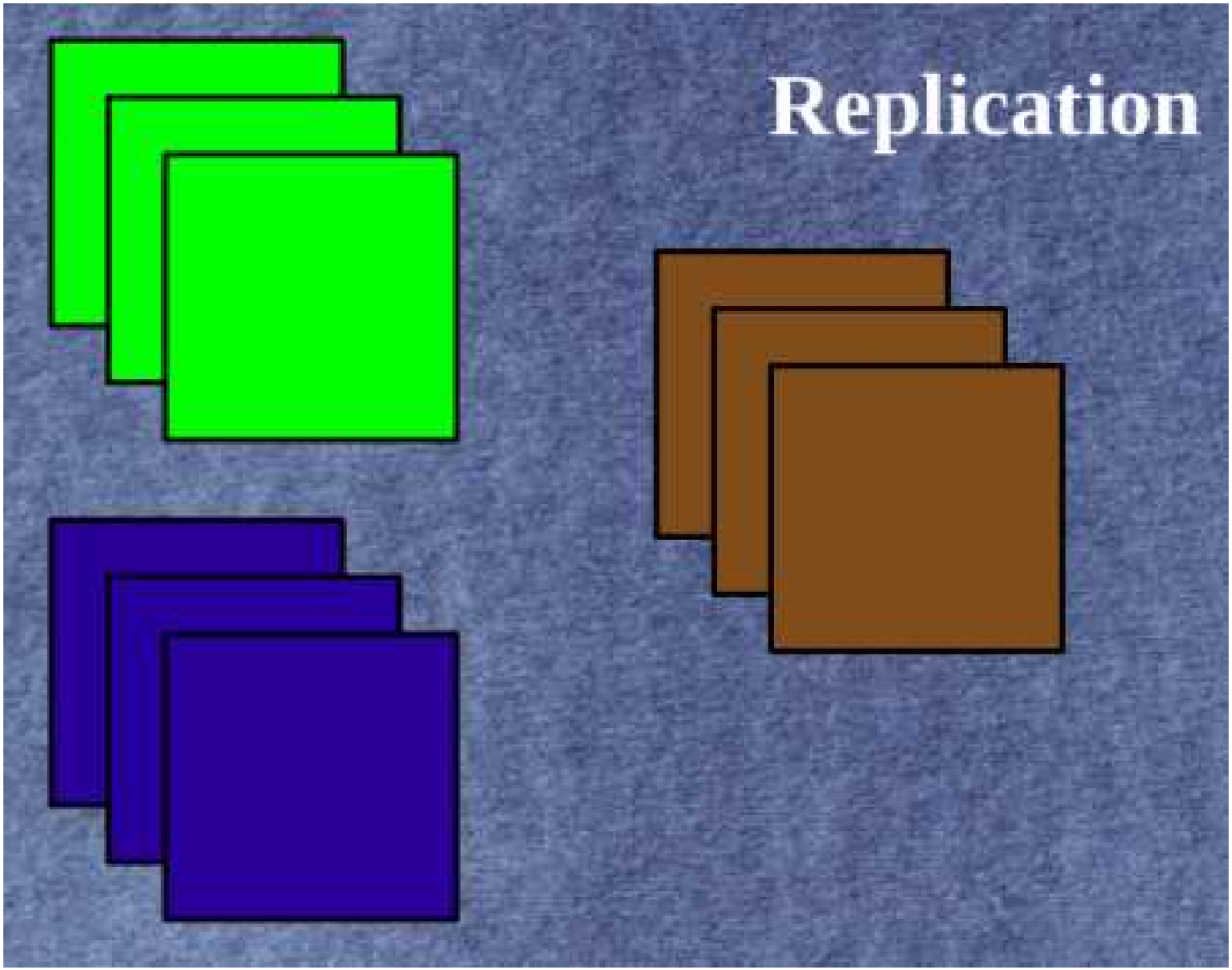
Block Placement

- Blocks – Default size 64MB
- Replica

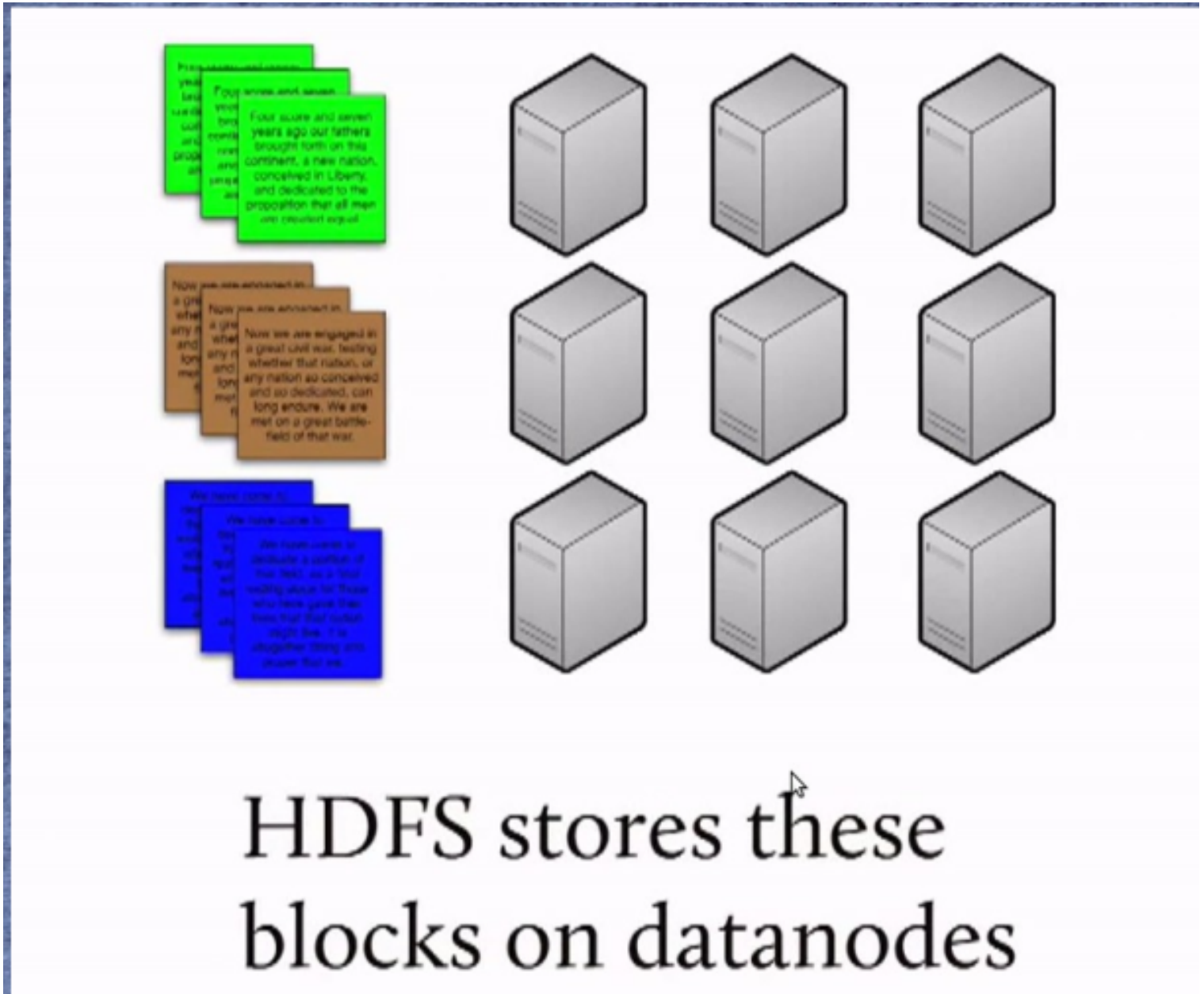
HDFS


This is our file
64MB
150MB
64MB
22MB
HDFS Splits it into Blocks

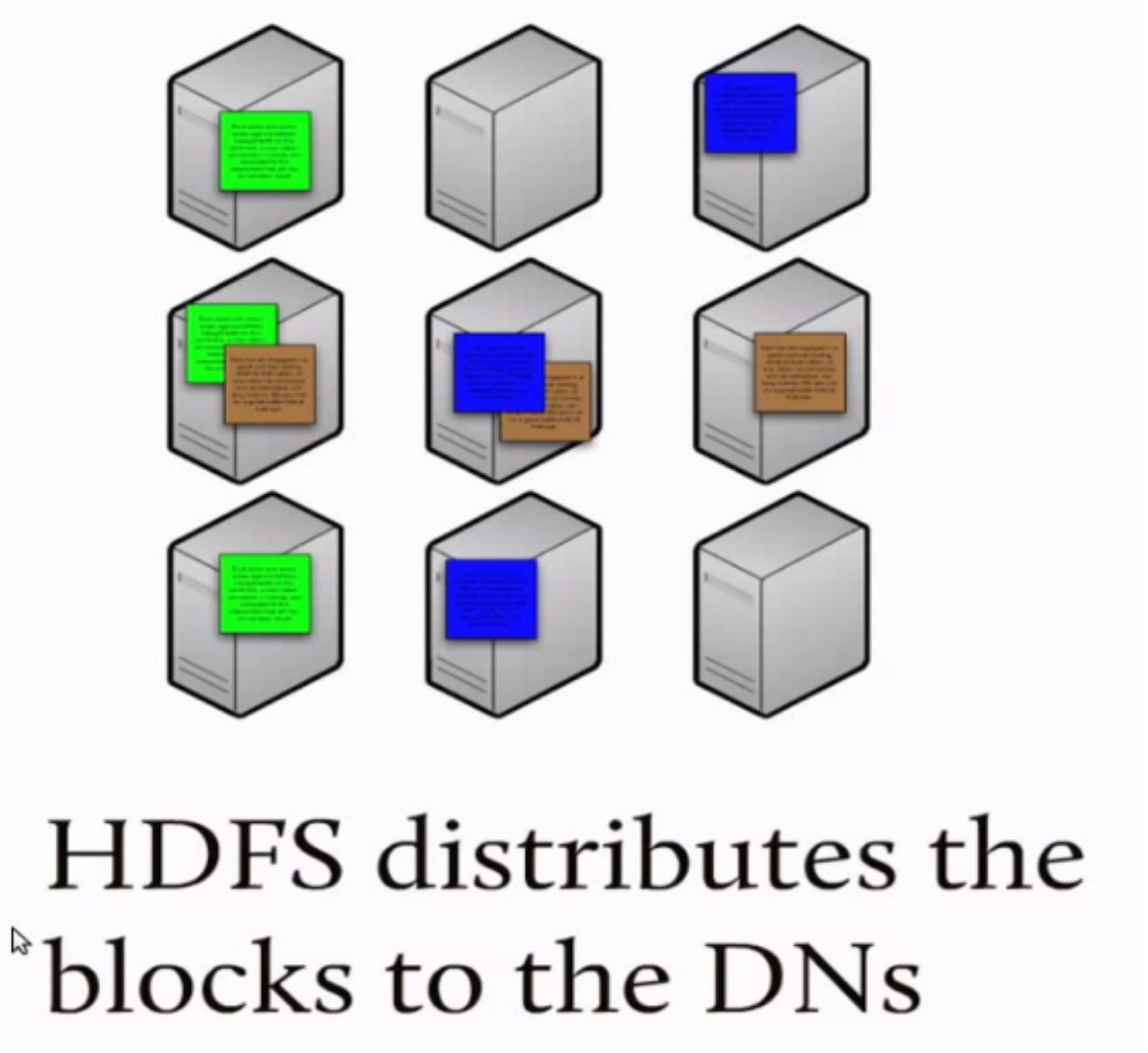
HDFS



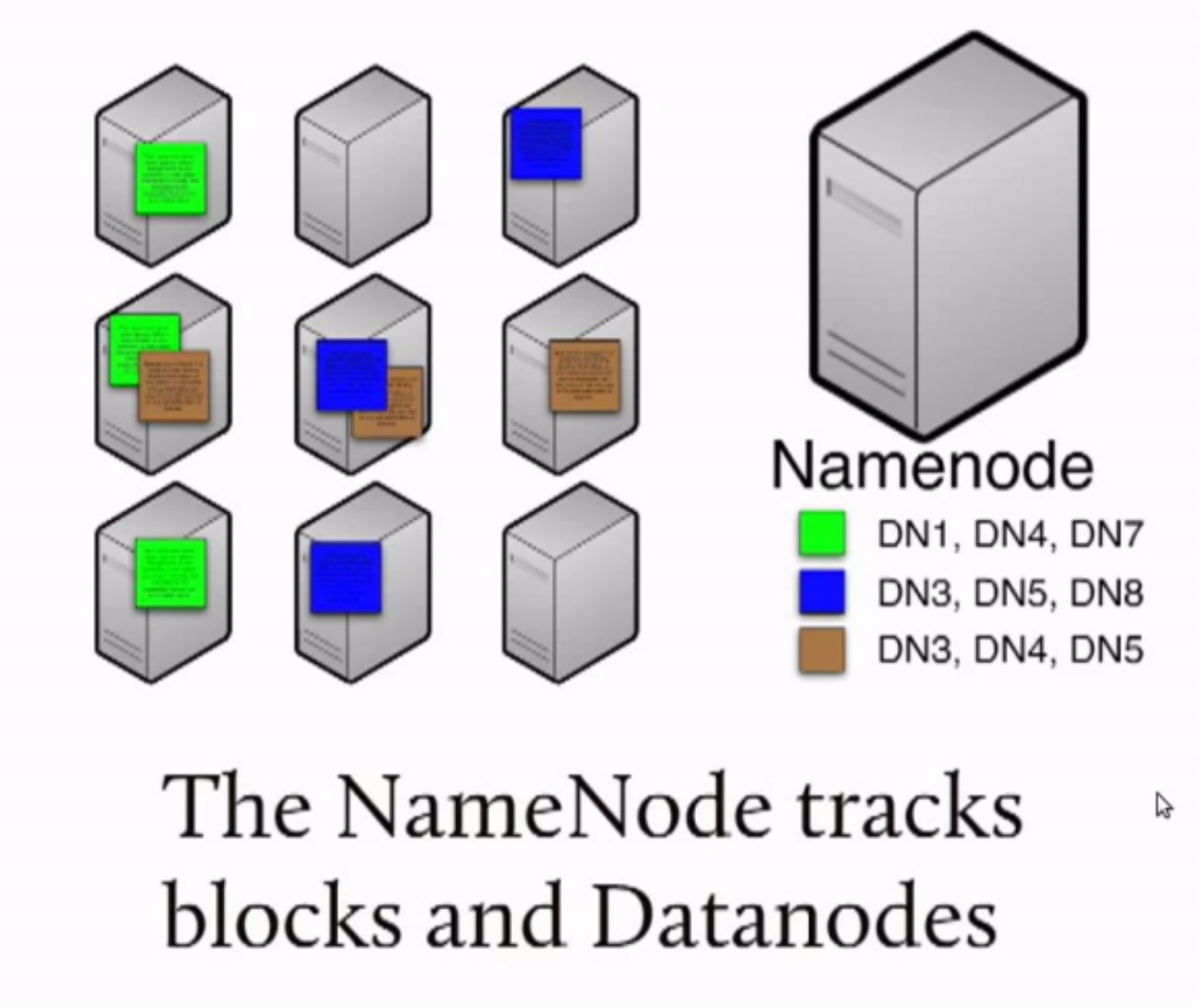
HDFS



HDFS



HDFS


Namenode
DN1, DN4, DN7
DN3, DN5, DN8
DN3, DN4, DN5
The NameNode tracks
blocks and Datanodes

Basic File system Operations

- hadoop fs -copyFromLocal input/docs/quangle.txt hdfs://localhost/user/tom/quangle.txt
- hadoop fs -copyToLocal quangle.txt quangle.copy.txt
- hadoop fs -mkdir books
- hadoop fs -ls
-

Hadoop Filesystems

- Local
- HDFS
- HFTP
- HSFTP
- WebHDFS
- HAR
- KFS (Cloud-Store)
- FTP
- S3 (native)
- S3 (block-based)
- Distributed RAID
- View

Filesystem -Local

- URI scheme -file
- Java implementation - fs.LocalFileSystem
- Description -A filesystem for a locally connected disk with client- side checksums. Use RawLocalFileSystem for a local filesystem with no checksums.

Filesystem - HDFS

- URI scheme -hdfs
- Java implementation - hdfs.DistributedFileSystem
- Description -Hadoop's distributed filesystem. HDFS is designed to work efficiently in conjunction with MapReduce.

Filesystem -HFTP

- URI scheme -hftp
- Java implementation - hdfs.HftpFileSystem
- Description -A filesystem providing read-only access to HDFS over HTTP. (Despite its name, HFTP has no connection with FTP.) Often used with distcp to copy data between HDFS clusters running different versions.

Filesystem - HSFTP

- URI scheme -hsftp
- Java implementation -hdfs.HsftpFileSystem
- Description -A filesystem providing read-only access to HDFS over HTTPS. (Again, this has no connection with FTP.)

Filesystem -WebHDFS

- URI scheme -webhdfs
- Java implementation -hdfs.web.WebHdfsFile System
- Description -A filesystem providing secure read-write access to HDFS over HTTP. WebHDFS is intended as a replacement for HFTP and HSFTP.

Filesystem - HAR

- URI scheme -har
- Java implementation -fs.HarFileSystem
- Description - A filesystem layered on another filesystem for archiving files. Hadoop Archives are typically used for archiving files in HDFS to reduce the namenode's memory usage.

Filesystem -KFS (Cloud-Store)

- URI scheme -kfs
- Java implementation -fs.kfs.KosmosFileSystem
- Description - CloudStore (formerly Kosmos filesystem) is a dis- tributed filesystem like HDFS or Google's GFS, written in C++

Filesystem - FTP

- URI scheme -ftp
- Java implementation -fs.ftp.FTPFileSystem
- Description - A filesystem backed by an FTP server.

Filesystem - S3 (native)

- URI scheme -s3n
- Java implementation -fs.s3native.NativeS3FileSystem
- Description - A filesystem backed by Amazon S3

Filesystem - S3 (block-based)

- URI scheme -s3
- Java implementation -fs.s3.S3FileSystem
- Description -A filesystem backed by Amazon S3, which stores files in blocks (much like HDFS) to overcome S3's 5 GB file size limit.

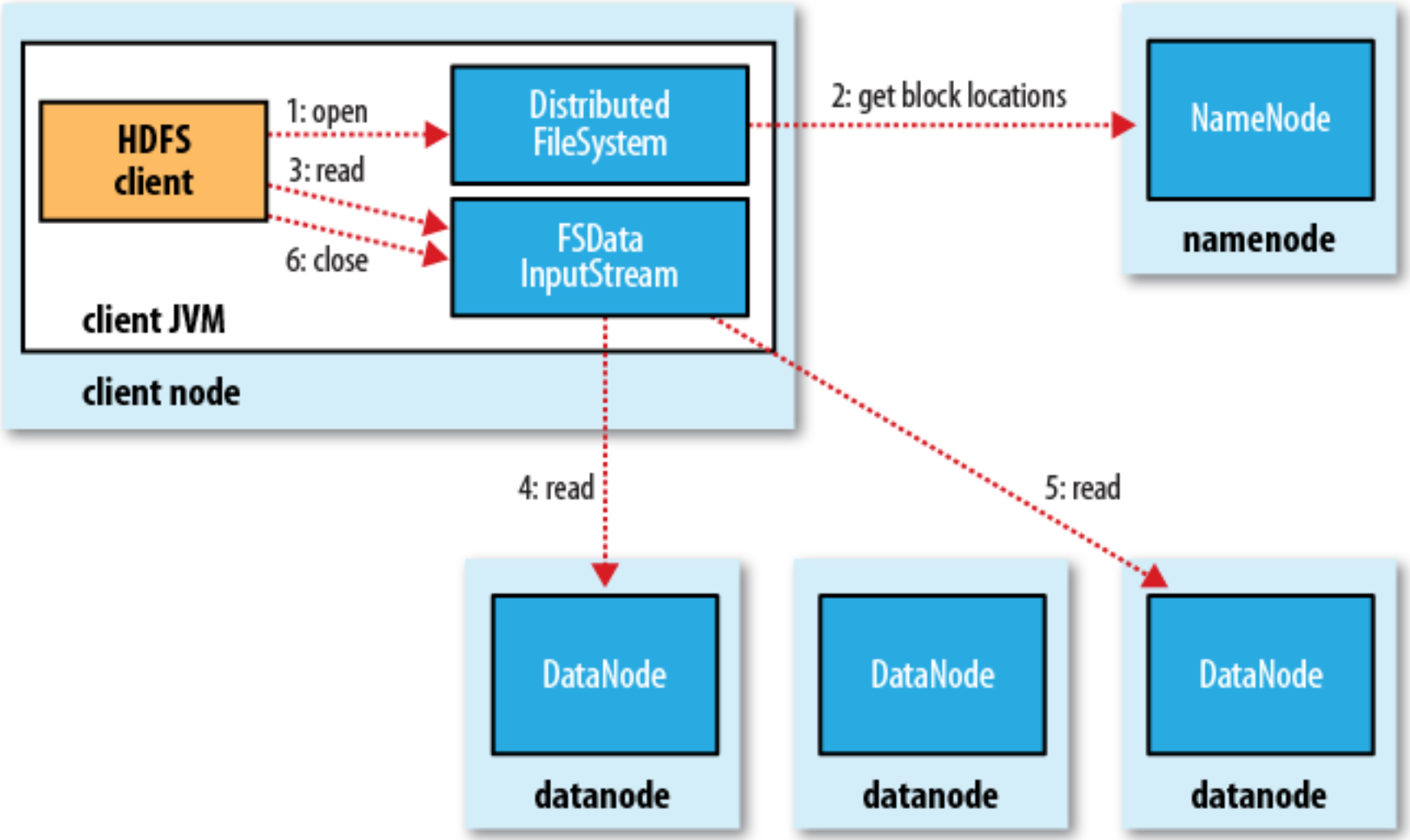
Filesystem - Distributed RAID

- URI scheme -hdfs
- Java implementation -hdfs.DistributedRaidFileSystem
- Description - A "RAID" version of HDFS designed for archival storage. For each file in HDFS, a (smaller) parity file is created, which allows the HDFS replication to be reduced from three to two, which reduces disk usage by 25% to 30% while keeping the probability of data loss the same. Distributed RAID requires that you run a RaidNode daemon on the cluster.

Filesystem - View

- URI scheme -viewfs
- Java implementation -viewfs.ViewFileSystem
- Description -A client-side mount table for other Hadoop filesystems. Commonly used to create mount points for federated namenodes

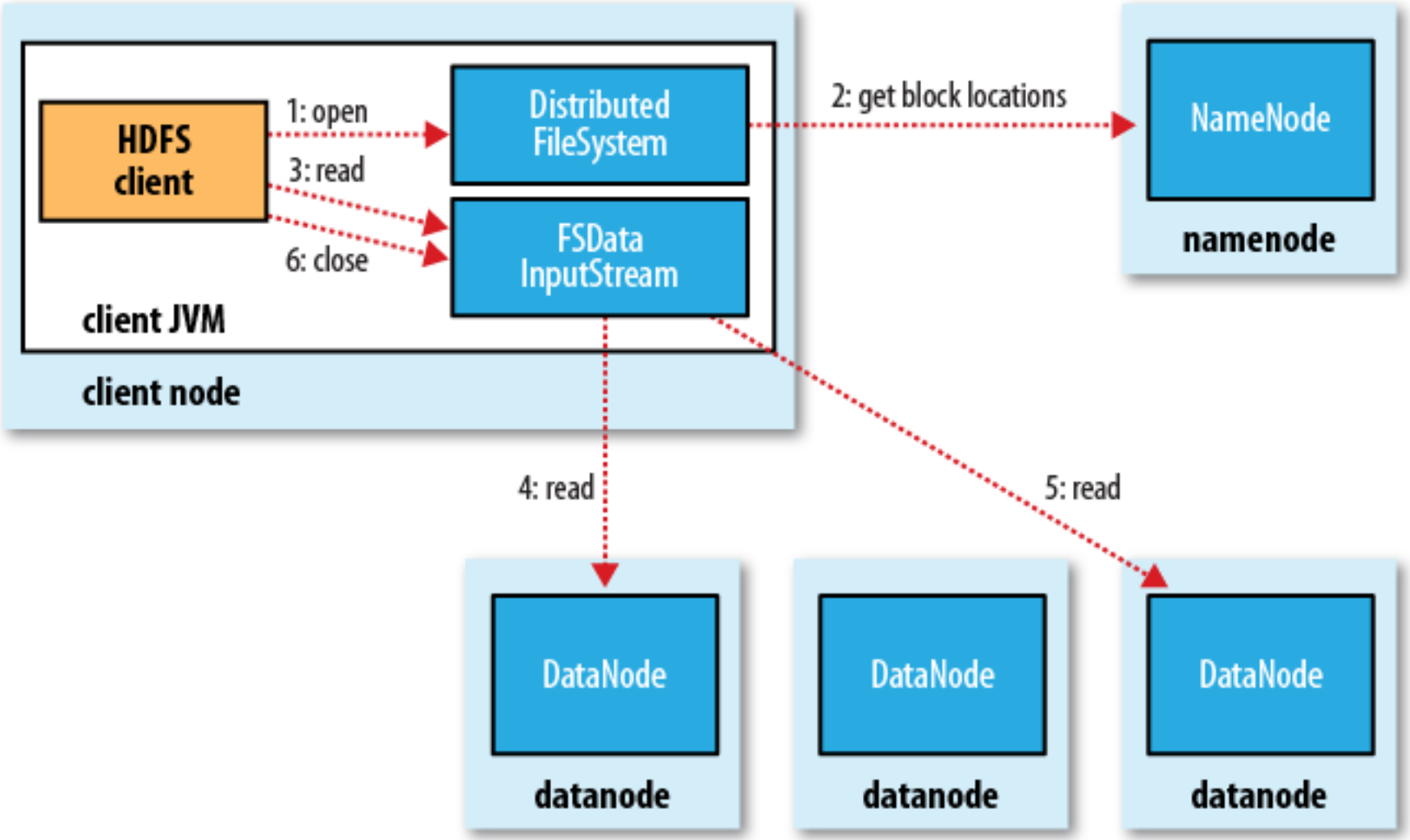
Anatomy of a File Read



HDFS-File Read

- HDFS client sends Read() to File system
- DFS - getblocklocation - Namenode
- client- read()-FSdata input stream
- Read data from Datanodes
- if Datanode down - fetch from next source- Remember the node
- after finishing the set , repeat the steps for moreblocks
- DFSInputStream verify the checksums- if error report to namenode
- Namenode- Inmemory computing

Anatomy of a file write



Replica Placement

- trade-off between reliability and write bandwidth and read bandwidth

Parallel Copying with distcp

Hadoop comes with a useful program called distcp for copying data to and from Hadoop filesystems in parallel.

- hadoop distcp file1 file2
- hadoop distcp dir1 dir2
- hadoop distcp -update dir1 dir2

Keeping an HDFS Cluster Balanced

- balancer tool-subsequently even out the block distribution across the cluster.