Introduction to HDFS and MapR-FS

A Few Quotes to Start Us Off

Data is a precious thing and will last longer than the systems themselves.

(Tim Berners-Lee, 2008)

Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom.

(Sir Arthur Conan Doyle)

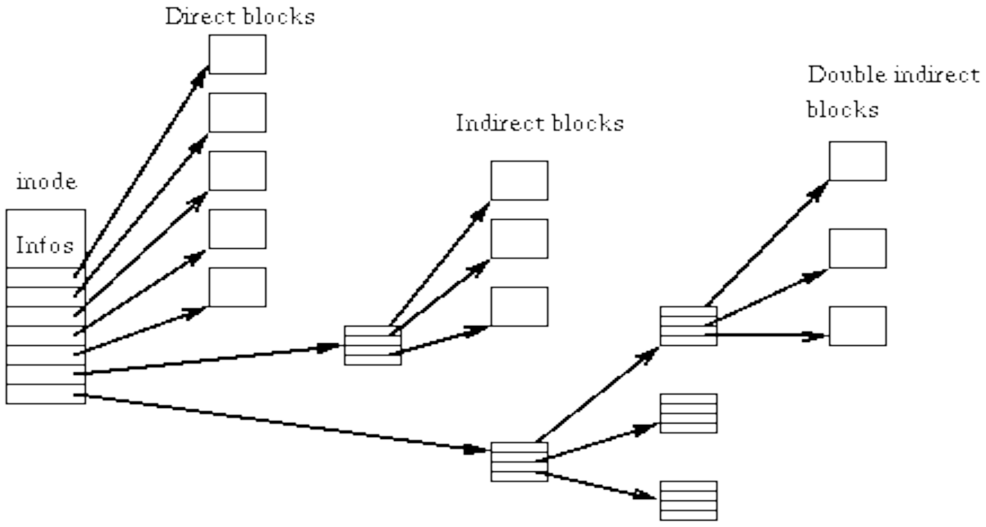
Learning Objectives

- Describe basic file system concepts
- Discuss an overview of HDFS
- Discuss an overview of MapR-FS
- Use the CLI to manage storage

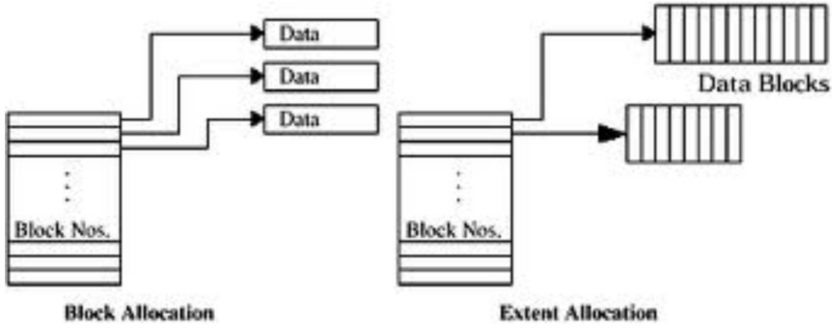
Describe Basic File System Concepts

Describe File System Concepts

- Logical structure that organizes files on a storage medium
- Dictates how data is stored and retrieved
- Contains data and metadata



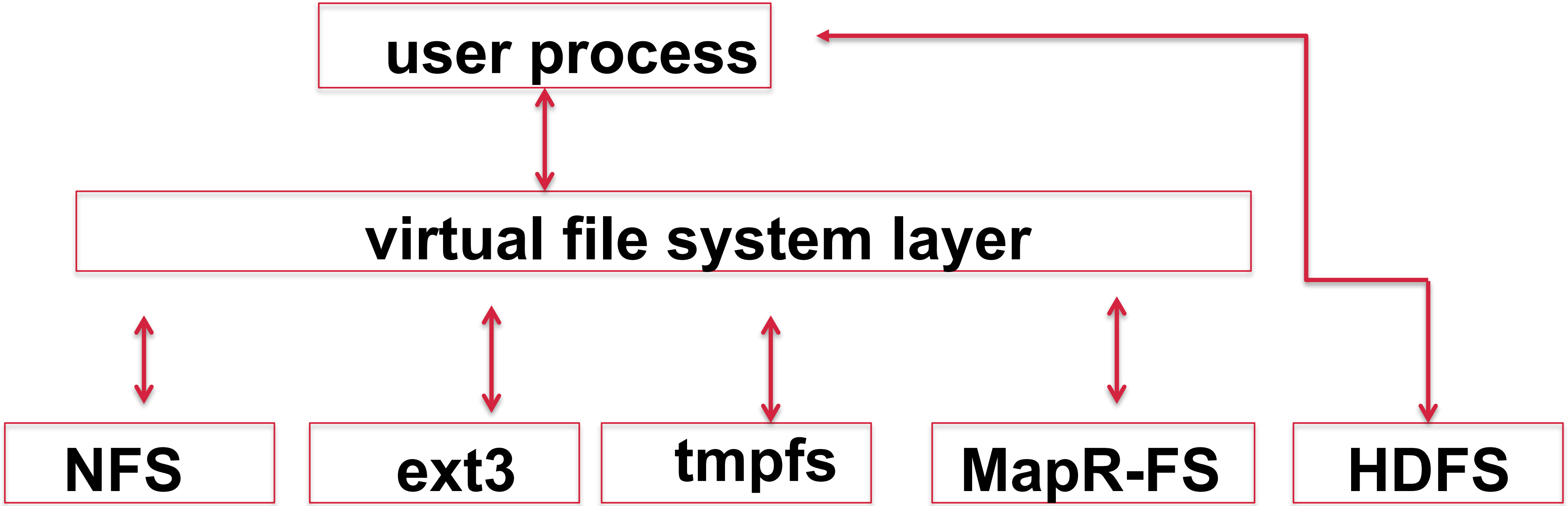
ext2 file system



block vs extent-based allocation

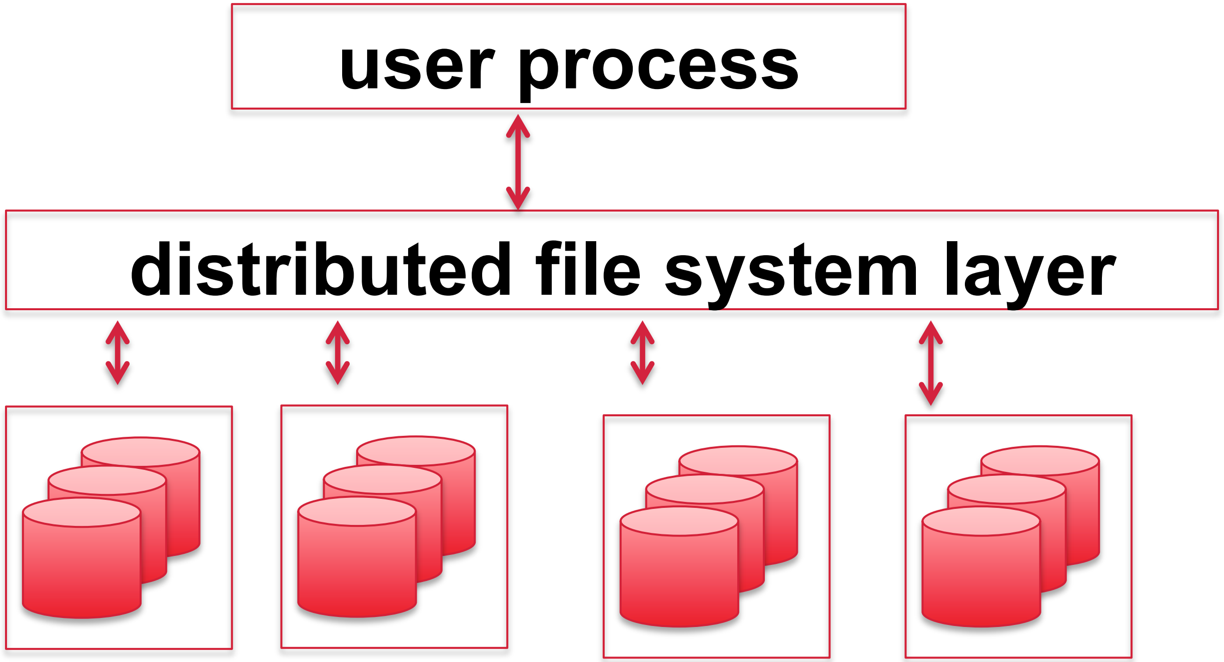
Describe the Purpose of a Virtual File System

- Translation layer from generic file system to real file system
- Enables standard POSIX file access



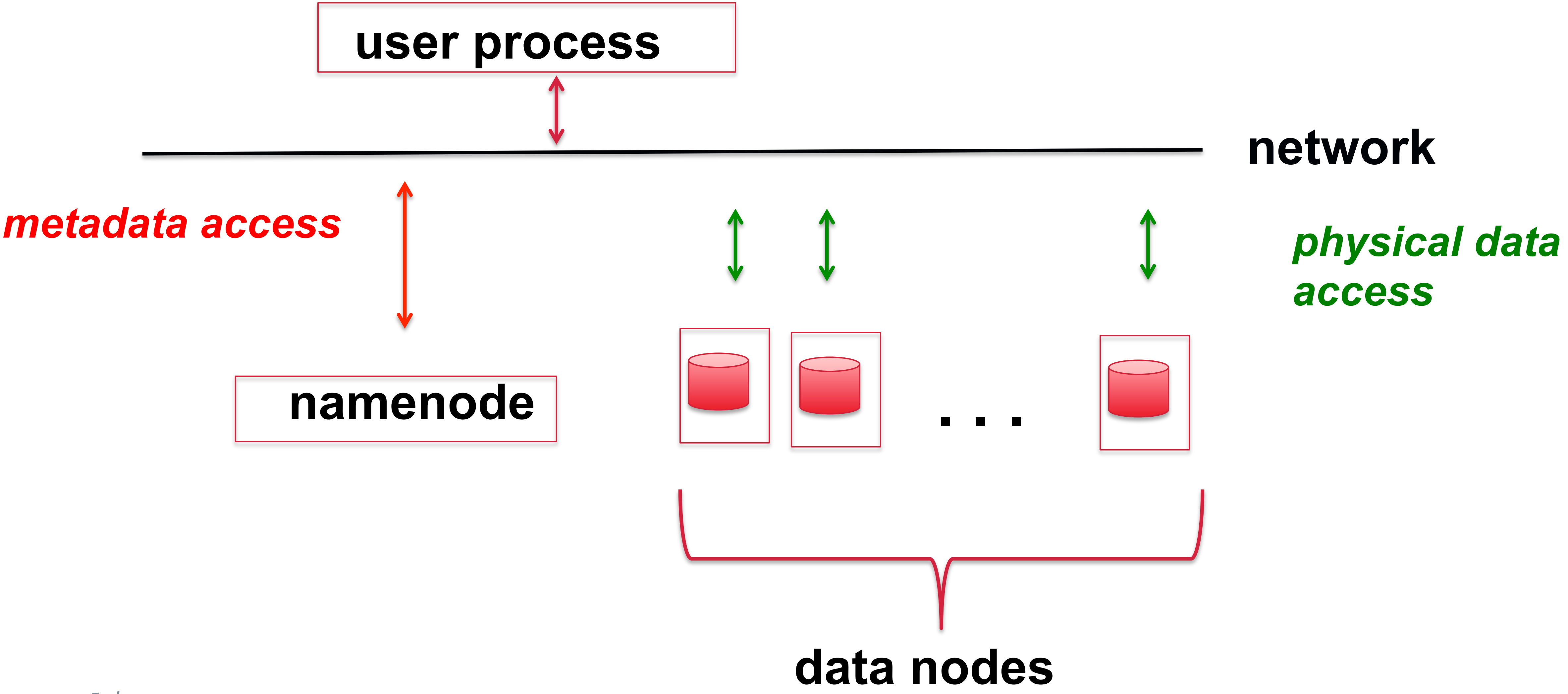
Describe Distributed File System Concepts

- Centrally stores metadata and distributes actual data
- Overcomes space, performance, and availability limitations of a single machine
- Abstracts data locality from client access

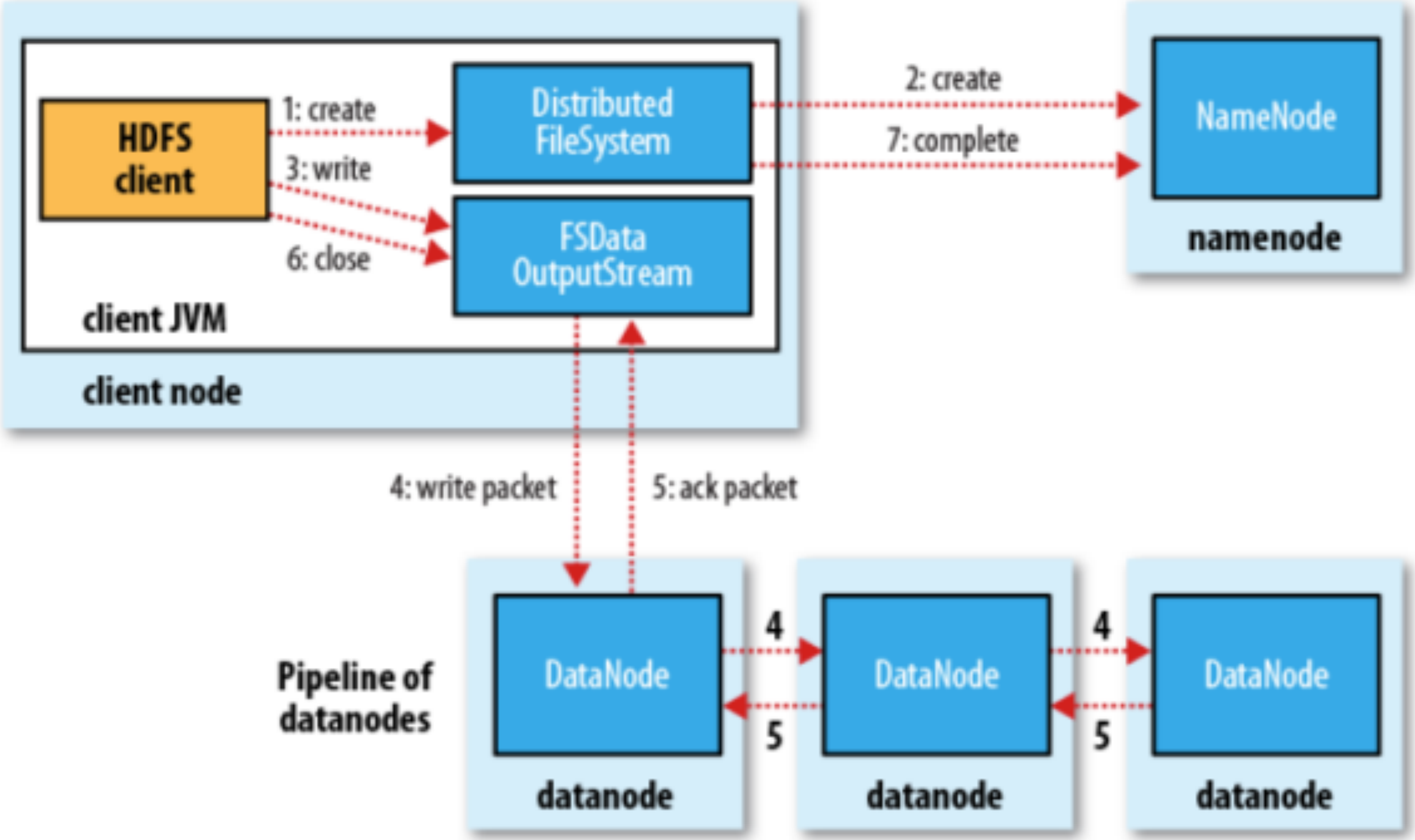


Discuss an Overview of HDFS

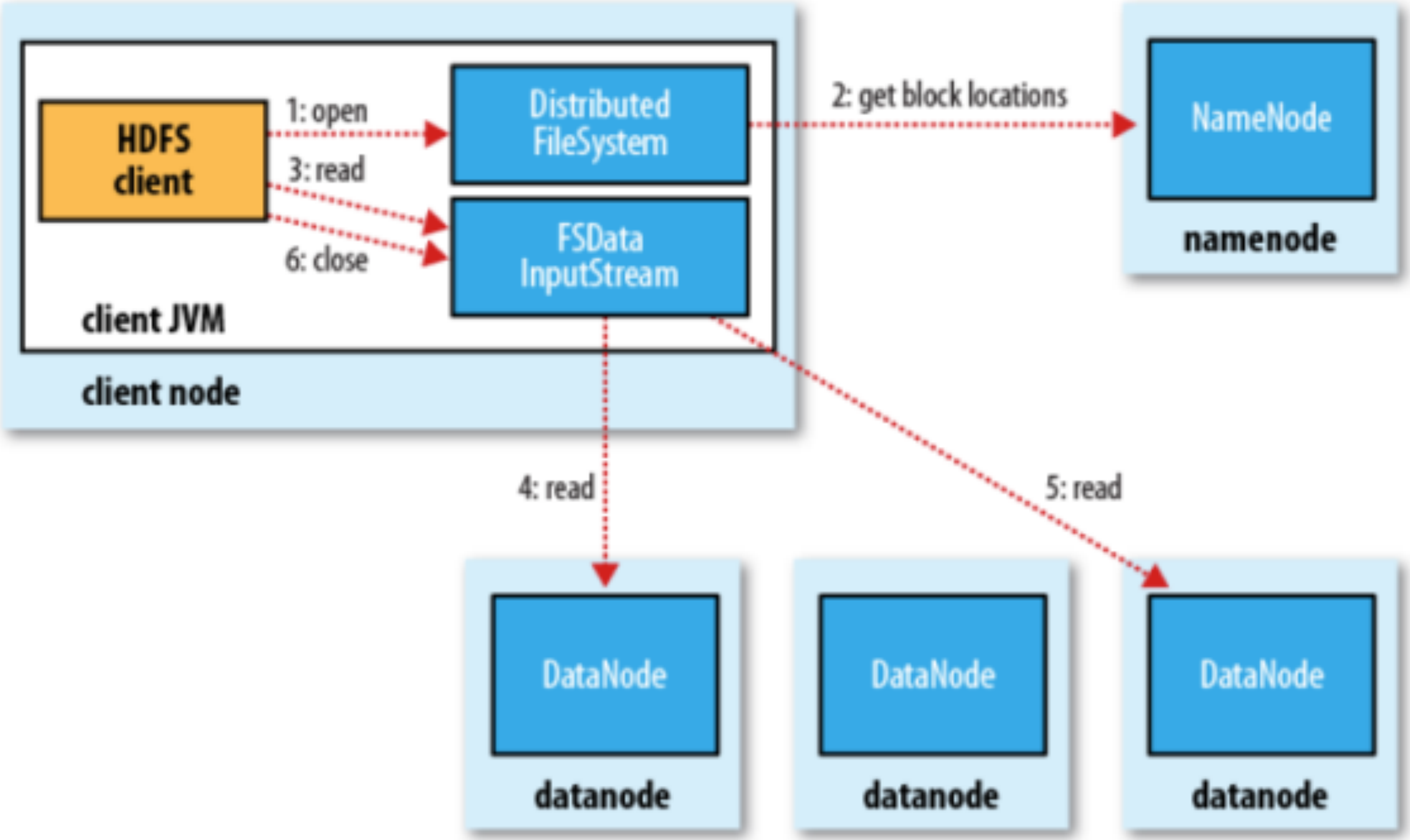
Describe the High-level HDFS Architecture



Discuss How Files are Written in HDFS



Discuss How Files are Read in HDFS

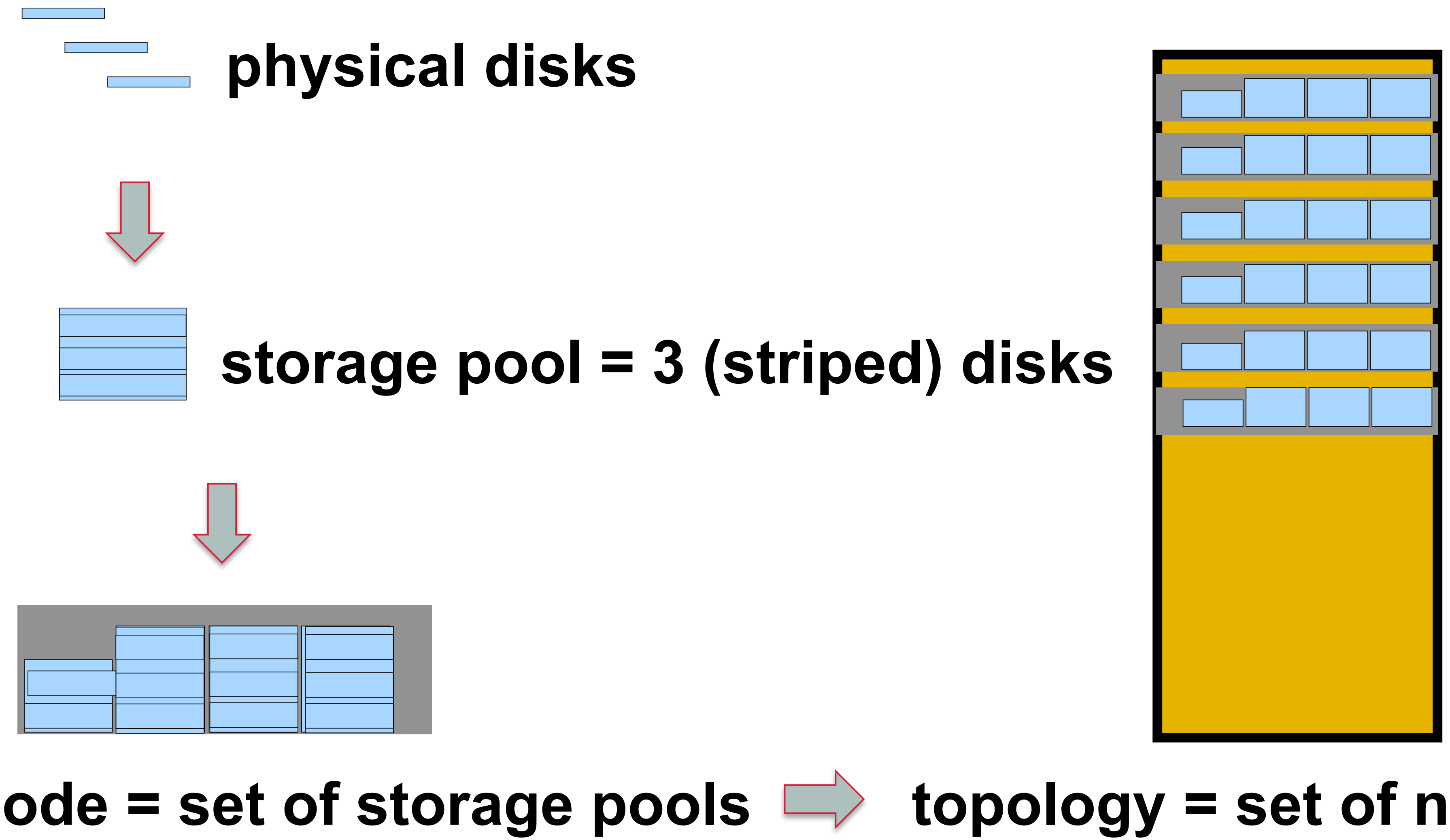


Identify Limitations of HDFS

aspect	limitation
Block size	Same size used for I/O, replication, and sharding
Mutability	Write-once, read-many
POSIX semantics	Must use ‘`hadoop fs`’ to access data
Availability	No snapshot or built-in mirroring capability
Scalability	Namenode only scales to 100M files
Performance	Written in Java and runs on block device

Discuss an Overview of MapR-FS

Describe the MapR Storage Framework



physical disks

storage pool = 3 (striped) disks

node = set of storage pools

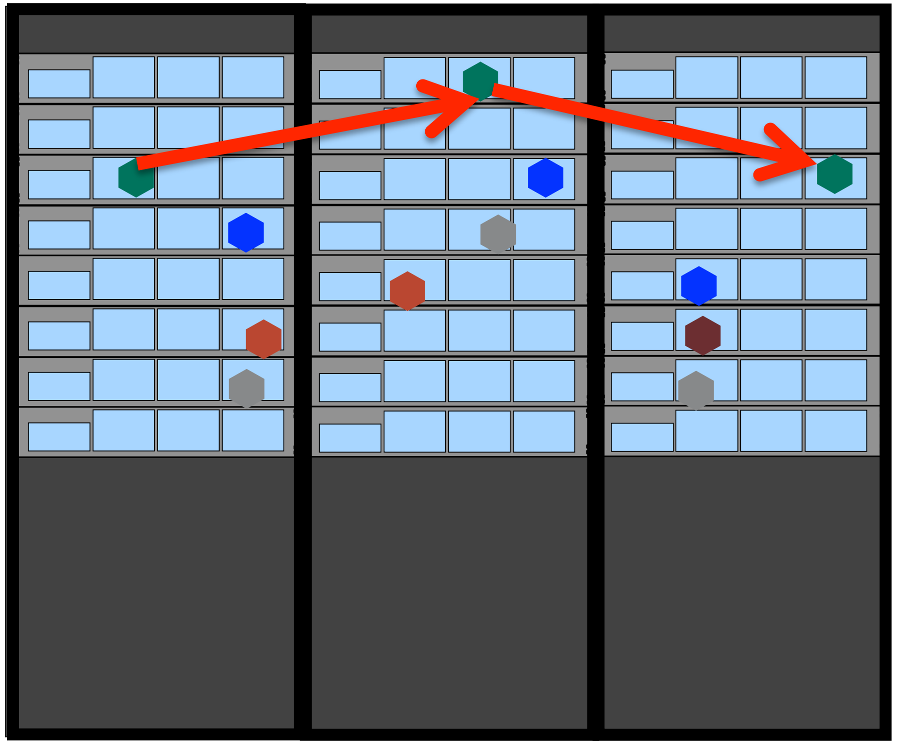
topology = set of nodes

Define Storage Pools and Containers

Data is written to *containers*

Containers are replicated

storage pool has many containers

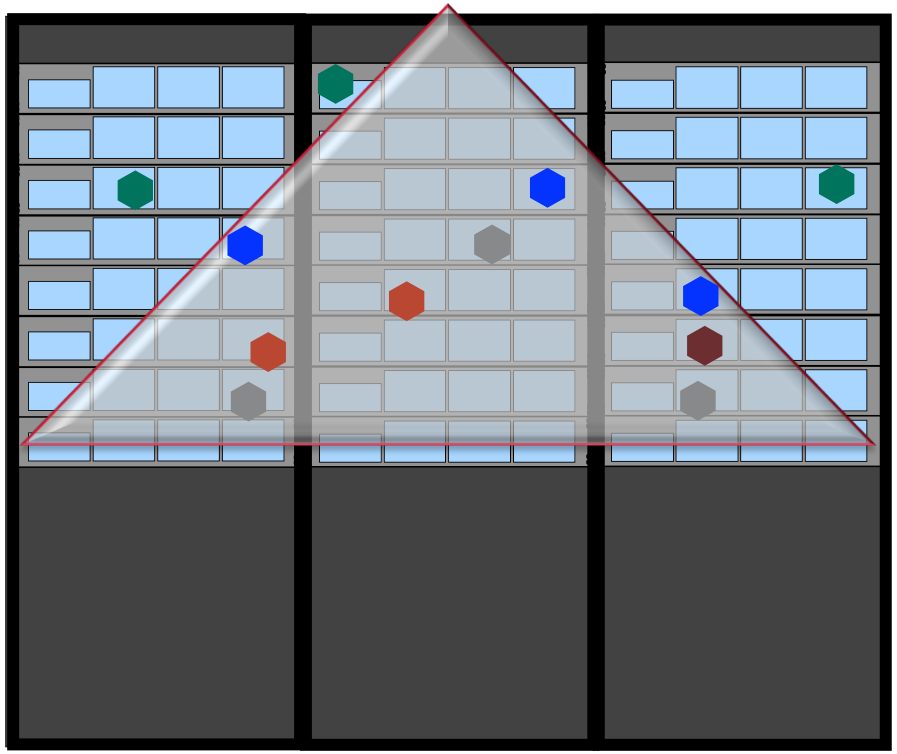


Define Containers and Volumes

Volumes are MapR specific features

Volumes group data containers across the cluster

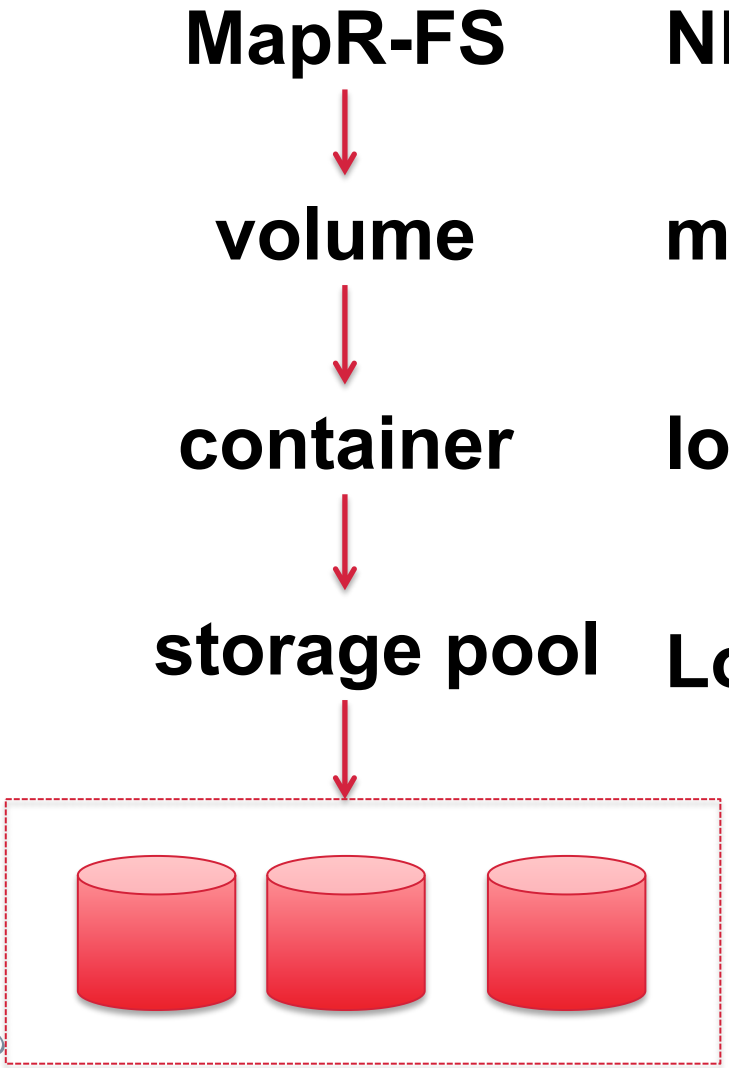
All replicas of a container are included in the same volume.



Identify MapR-FS Volume Features

Feature	Description
topologies	Provide placement policies for data
compression	Compresses data as it is being written to disk
mirrors	Copy data locally or remotely for protection
snapshots	Maintain point-in-time data and updates
quotas	Restrict total capacity per-user or per-group
permissions	Restrict access to users and groups
replication	Replicate containers in volume across cluster

Describe a Summary of Storage Structures



MapR-FS	NFS-exportable POSIX file system
volume	management abstraction
container	location/replication abstraction
storage pool	Logical-physical abstraction

Cite Differences b/w MapR-FS and HDFS

aspect	feature
Block size	Different sizes used for sharding, replicating, and performing I/O
Mutability	Full read-write capability
Access	Can NFS-mount MapR-FS volumes
POSIX semantics	Can use native OS to access data
Availability	Snapshots and local/remote mirroring support
Scalability	No limit to the number of files
Performance	Written in C and runs on raw device

Compare Block Sizes in MapR-FS and HDFS

Storage unit	HDFS	MapR-FS
Unit of sharding	Block = 64 MB	Chunk = 256 MB
Unit of replication	Block = 64 MB	Container = 16-32 GB
Unit of I/O	Block = 64 MB	Block = 8 KB

MapR-FS disambiguates use for each storage unit to optimize use case

Using the CLI to Manage Data

Use the `hadoop fs` CLI

***Usage:** `hadoop fs [command] [args]`*

```
hadoop fs -mkdir mydir

hadoop fs -copyFromLocal /etc/hosts mydir

hadoop fs -lsr mydir

hadoop fs -cat mydir/hosts

hadoop fs -rm mydir/hosts
```

Differentiate Absolute and Relative Paths

```
$ hadoop fs -ls /
data1 data2 tmp user var
```

```
$ hadoop conf -dump | grep fs.default.name
fs.default.name=maprfs:///
```

```
$ hadoop fs -ls
/user/jcasaletto/IN /user/jcasaletto/OUT
```

```
$ hadoop conf -dump | grep fs.mapr.working.dir
fs.mapr.working.dir=/user/$USERNAME/
```

Use the `hadoop mfs` CLI

Usage: hadoop mfs [command] [args]

```
hadoop mfs -ln mydir yourdir

hadoop mfs -setcompression off mydir

hadoop mfs -setchunksize 65536 mydir
```

Use the Operating System CLI

```
mkdir /user/james/mydir

cp /etc/hosts /user/james/mydir

ls -R /user/james/mydir

ln -s /user/james/mydir /user/james/yourdir

rm  /user/james/mydir/hosts

tail / grep / awk / sed
```