Big Data
HDFS

K V Subramaniam
Computer Science and Engineering

Lecture overview - HDFS

- Need for Distributed file systems
- HDFS Introduction
- Architecture
- Operations
- Internals
- Fault Tolerance and replication
- Blocks

Big Data: File systems and distributed file systems

Data Growth –Why the need for HDFS?

As per RBI in May 2019,

#credit/debit card transactions~ 1.3 Billion

(https://rbidocs.rbi.org.in/rdocs/ATM/PDFs/ATM052019E96EC259708C4ED9AD9E0C6B5E8B6DD5.PDF)

If each transaction requires about 10K of data

13 TB of data

Lot of data and this is only for credit/debit card transactions

There are other transactions also

Suppose you want to look for fraudulent transactions

How to store and process this data?

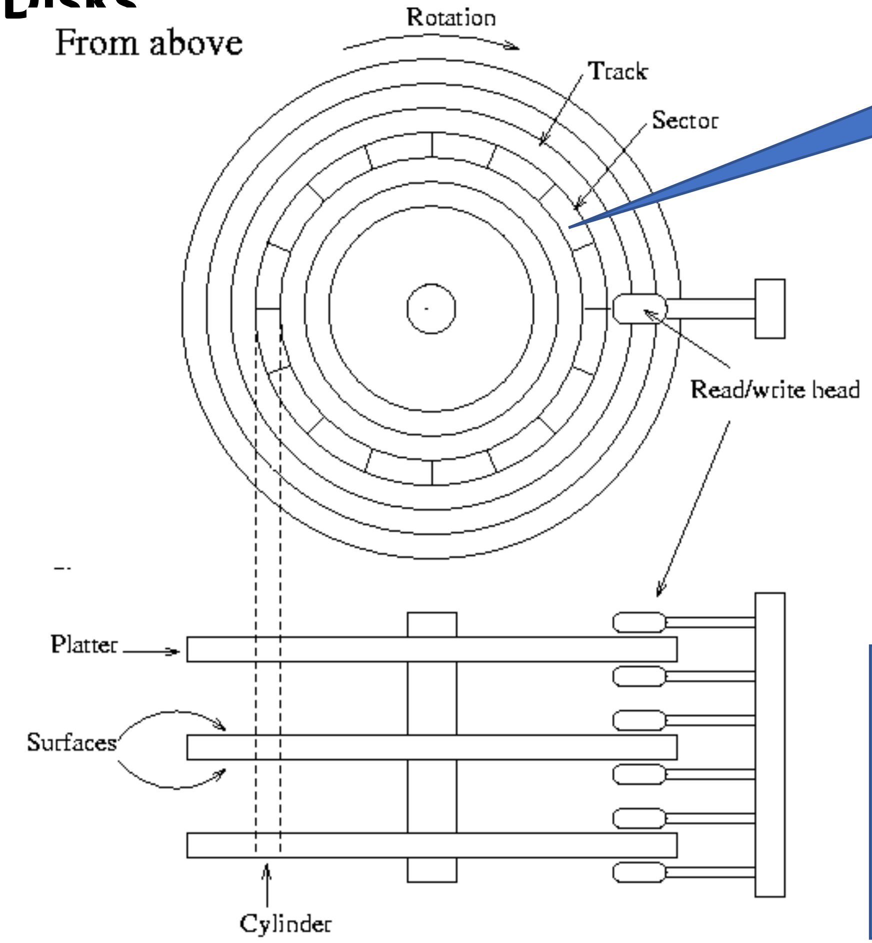
This class

FLOOD OF DATA BY 2020
AVG INTERNET USER 1.5 GB OF TRAFFIC/DAY
AUTONOMOUS VEHICLES 4.0 TB OF DATA/DAY*
CONNECTED AIRPLANE 5.0 TB OF DATA/DAY
SMART FACTORY 5.0 PB OF VIDEO/DAY

Image Courtesy: https://www.indiamart.com/proddetail/big-data-analytics-service-6764825662.html

Disk Storage –persistent storage

Persistent Storage - Disks



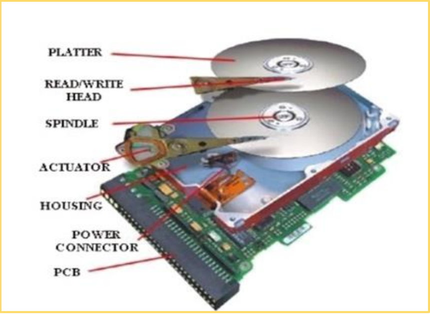
Block oriented device : storage divided into fixed size blocks

Can we store the persistent data directly on these blocks?

Who will maintain the meta-data?

Concerns: Which block contains the data as files are not necessarily multiples of blocks size

A filesystem layer on top of disk manages blocks and maps data/metadata to blocks



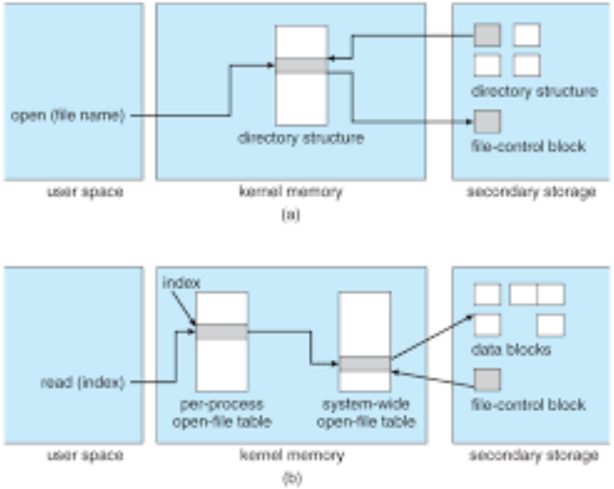
HDFS –Hadoop distributed File system

FILES

- Named collection of related information
- on disks

Desirable properties of files

- Long-term existence
- Sharable between processes
- Access permissions



https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/12_FileSystemImplementation.html

Distributed File System

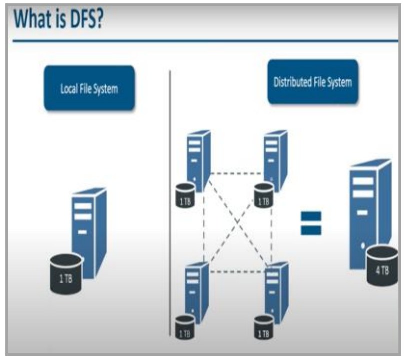
Consider case when data is so large that it cannot fit on a single disk.

- A DFS manages files and folders across multiple computers.

DFS can organize and display the files as if they are stored on one computer.

It serves the same purpose as a traditional file system.

Designed to provide file storage and controlled access to files over local and wide area networks.



Exercise

Consider that you have 1TB of data?
Compare the time taken to read data in both the cases below

- single machine (*4 I/O channels each channel 100mb/s)*
- 10 machines (*each having 4 I/O channels each channel 100mb/s*).

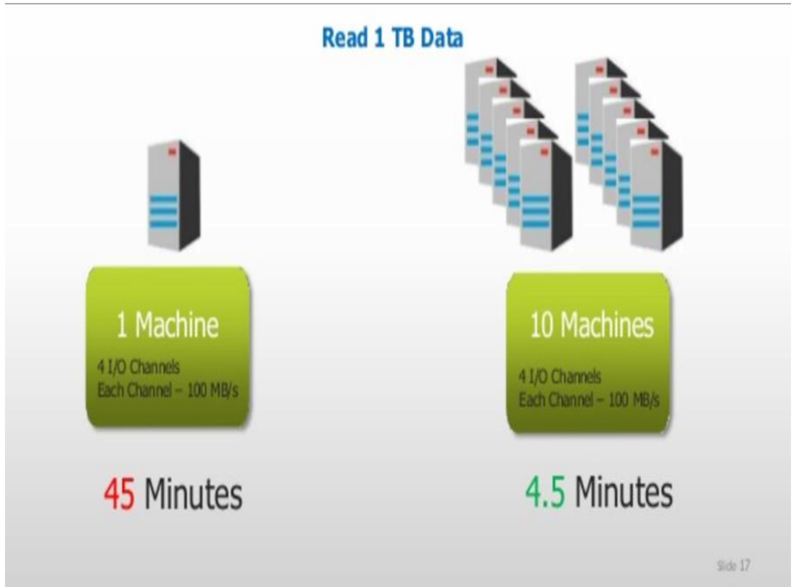
Big Data: HDFS Introduction

Exercise Solution

Consider that you have 1TB of data?

Compare the time taken in both the cases below

- single machine (*4 I/O channels each channel 100mb/s)*
- 10 machines (*each having 4 I/O channels each channel 100mb/s*).



HDFS –History

HDFS –Inspired by GFS

GFS –Google File System (2003)

Distributed File system on a cluster of machines

Developed by Doug Cutting and Mike Cafarella

Origin - Apache Nutch

- Goal : web search engine on 1 Billion Pages

Open source

HDFS –Hadoop distributed File system

“HDFS is a filesystem designed for storing very large files with streaming data access patterns, running on clusters of commodity hardware.”

Very large

- Files can be MB/GB/TB in size
- Hadoop clusters that are PB are currently operational

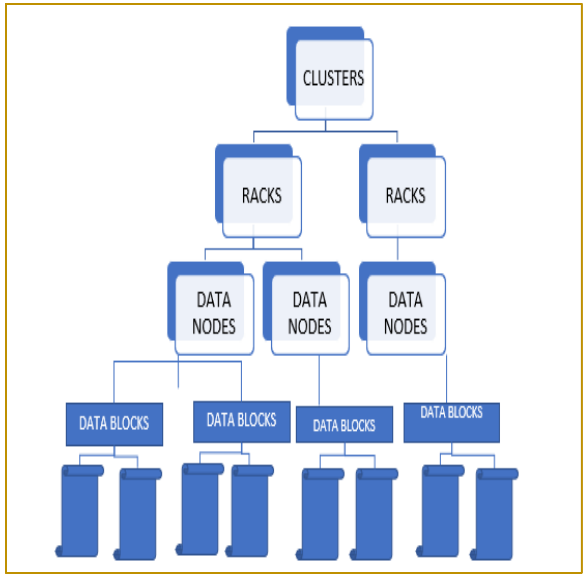
Read Mostly data

- most efficient data processing pattern is a write-once, read-many-times pattern.
- Each analysis will involve a large proportion of the dataset
- time to read the whole dataset is more important than the latency in reading the first record.

Commodity hardware

- Hadoop doesn’t require expensive, high reliable hardware
- Designed to run on clusters of commodity hardware

HDFS –Hadoop distributed File system



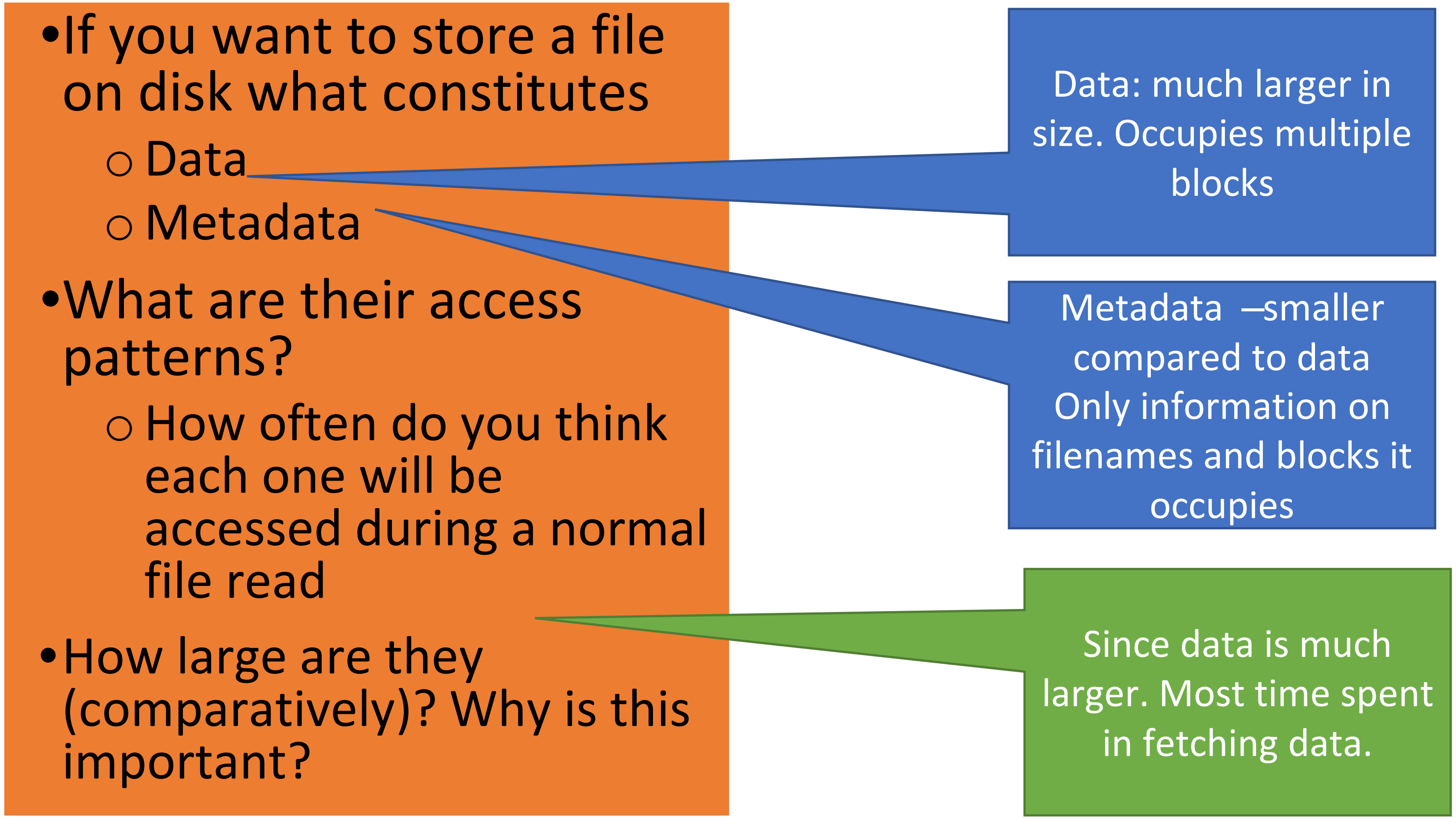
https://medium.com/@Meela349588204/rack-server-vs-blade-server-d1b5aee58ca2

Exercise

- If you want to store a file on disk what constitutes
  - Data
  - Metadata
- What are their access patterns?
  - How often do you think each one will be accessed during a normal file read
- How large are they (comparatively)? Why is this important?

Big Data: HDFS Architecture

Exercise Solution



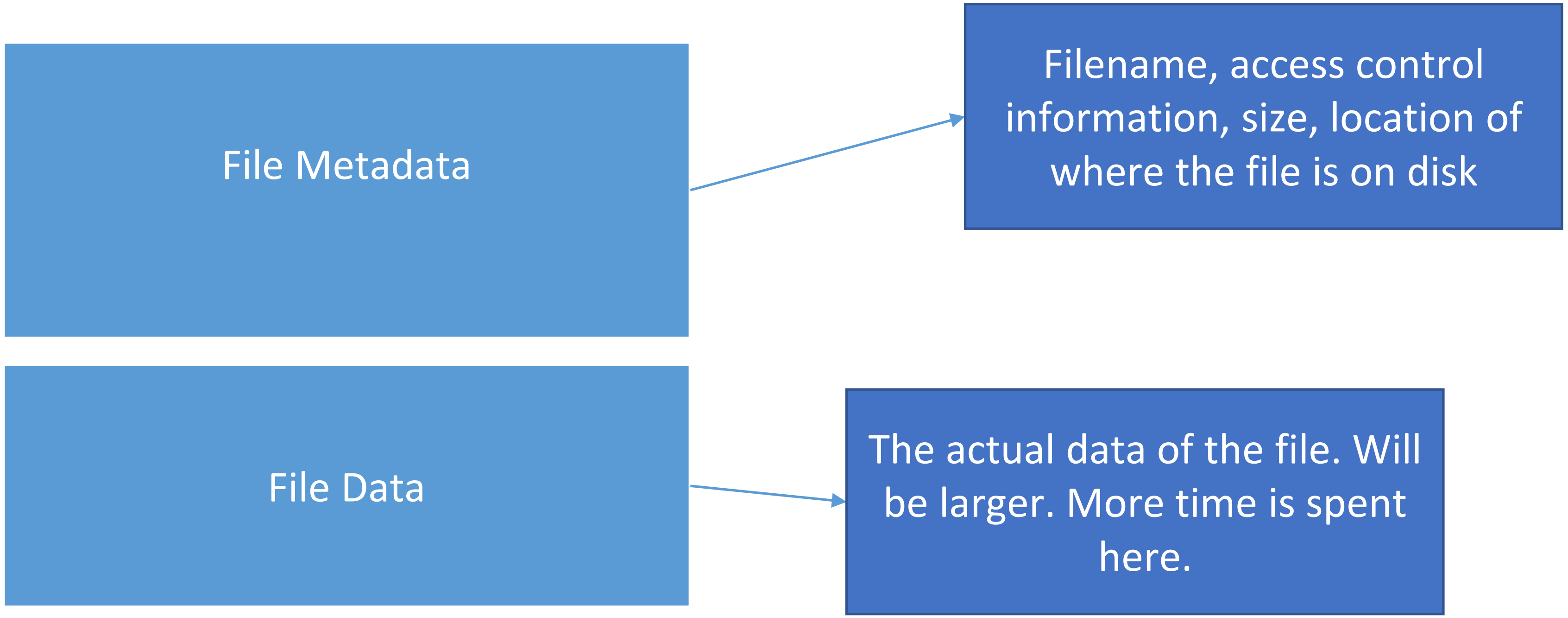
- If you want to store a file on disk what constitutes
  - Data
  - Metadata
- What are their access patterns?
  - How often do you think each one will be accessed during a normal file read
- How large are they (comparatively)? Why is this important?

Data: much larger in size. Occupies multiple blocks

Metadata –smaller compared to data Only information on filenames and blocks it occupies

Since data is much larger. Most time spent in fetching data.

HDFS –Architecture Motivation



File Metadata

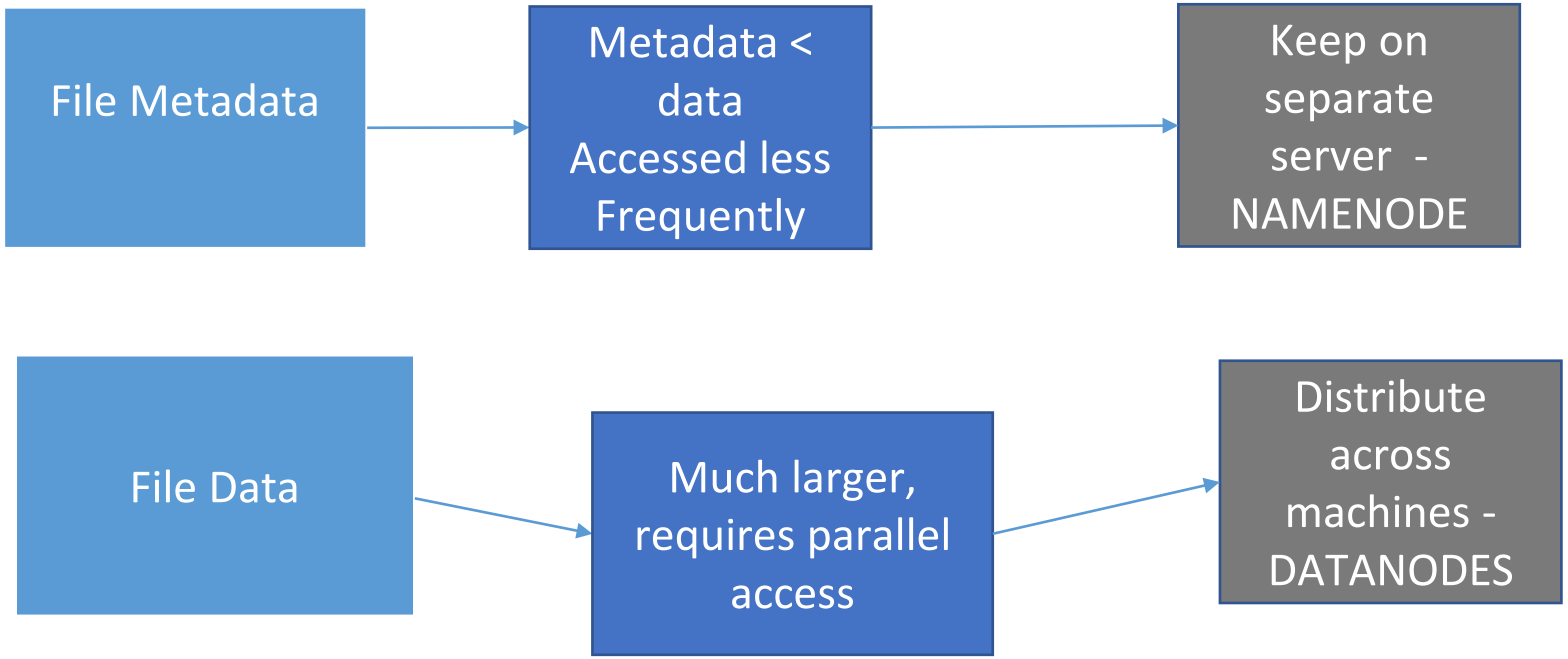
Filename, access control information, size, location of where the file is on disk

File Data

The actual data of the file. Will be larger. More time is spent here.

HDFS –Architecture Motivation

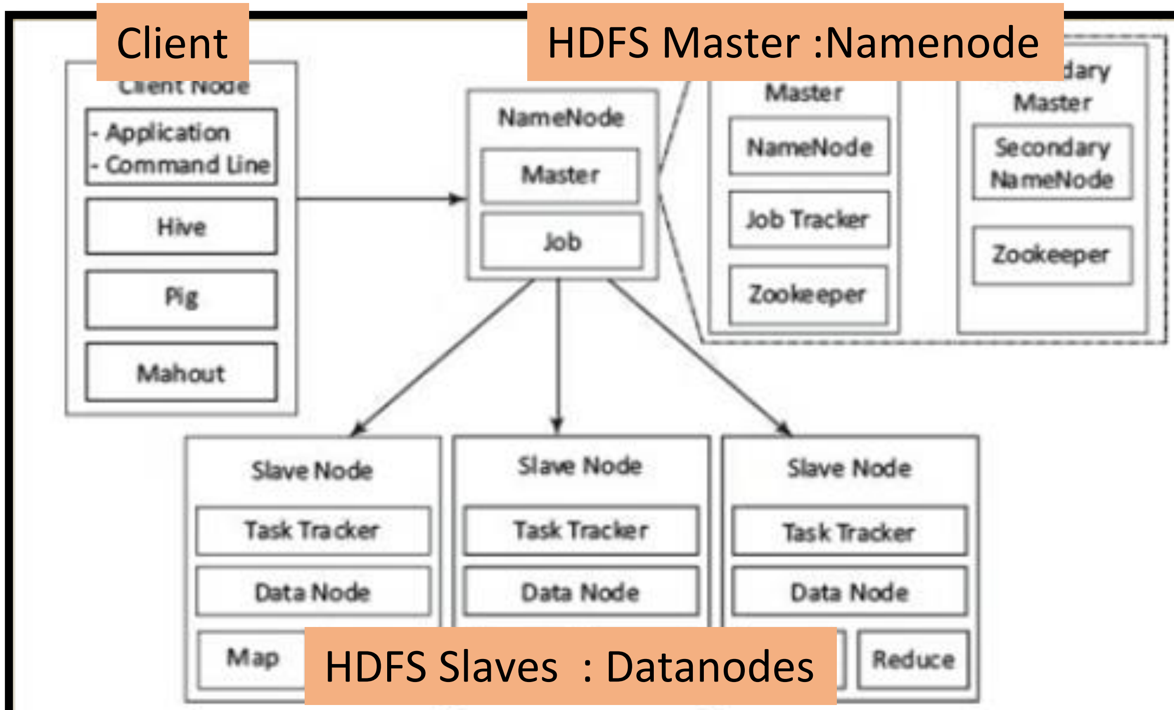
Solution



File Metadata → Metadata < data Accessed less Frequently → Keep on separate server - NAMENODE

File Data → Much larger, requires parallel access → Distribute across machines - DATANODES

HDFS –Master Slave Architecture



Client

HDFS Master :Namenode

HDFS Slaves : Datanodes

Figure 2.4 The client, master NameNode, MasterNodes and slave nodes

Diagram Courtesy: Big Data Analytics (Kindle Location 2502). McGraw Hill Education. Kindle Edition.

Big Data: HDFS Operations

BIG DATA

HDFS –Writing a file

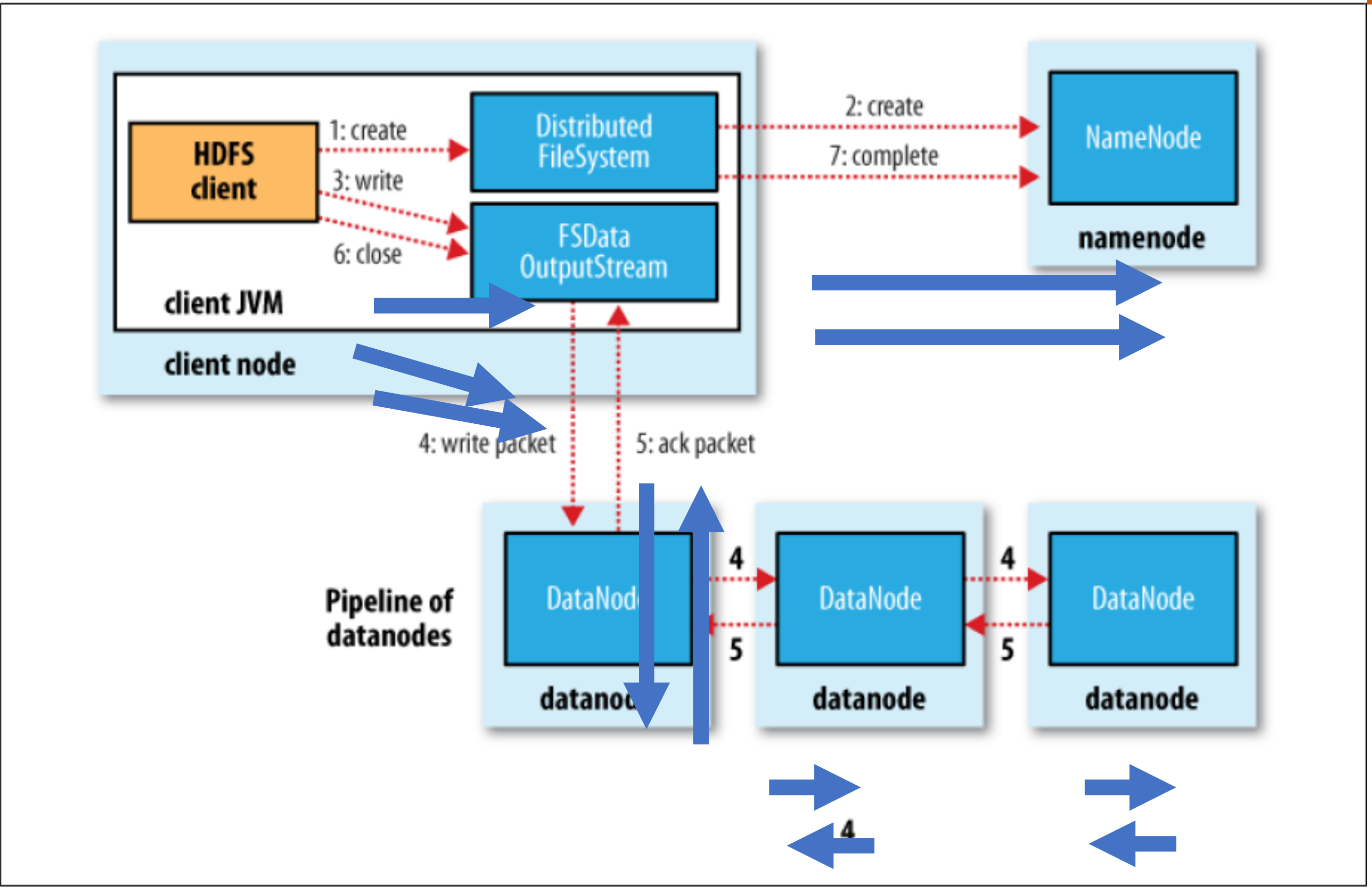
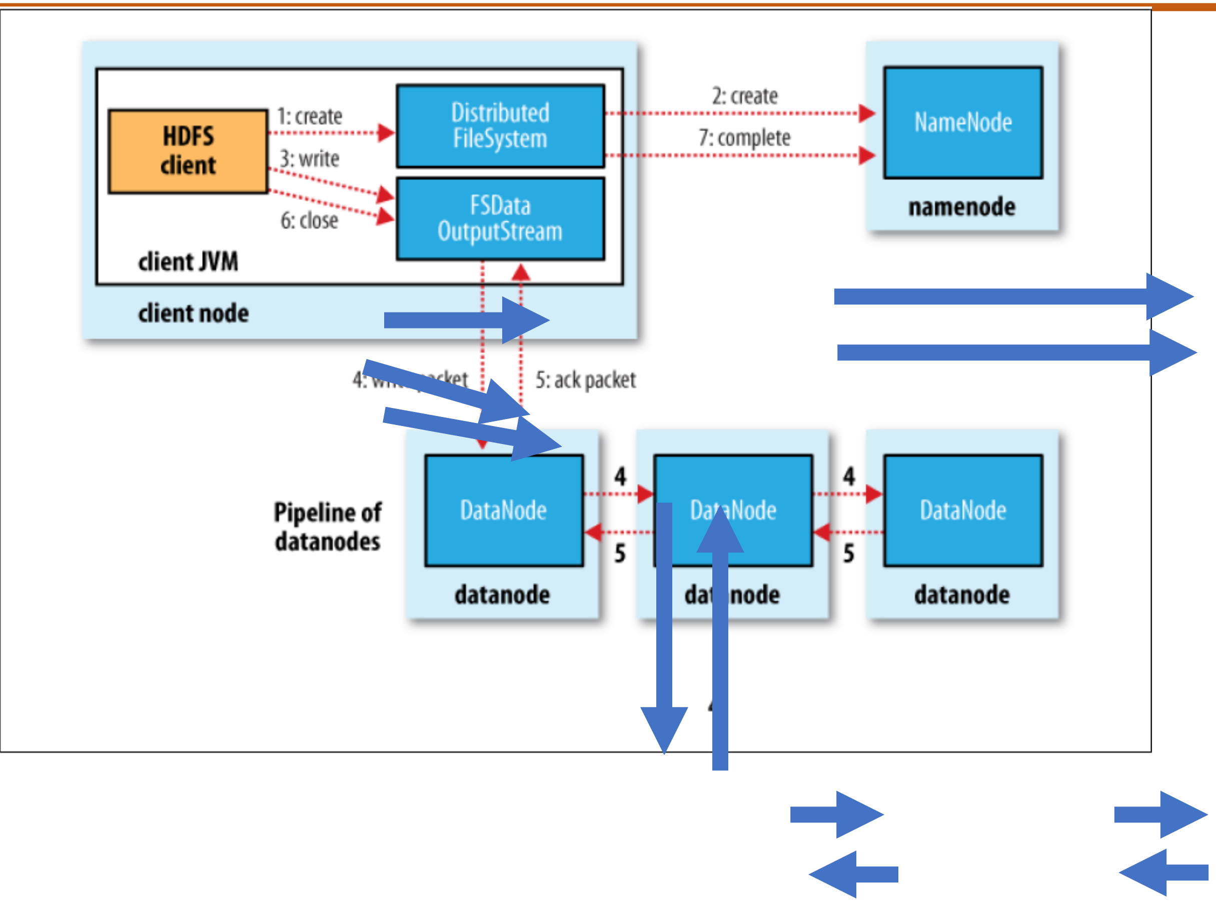


Image source: Hadoop Definitive Guide, Tom White, O'Reilly

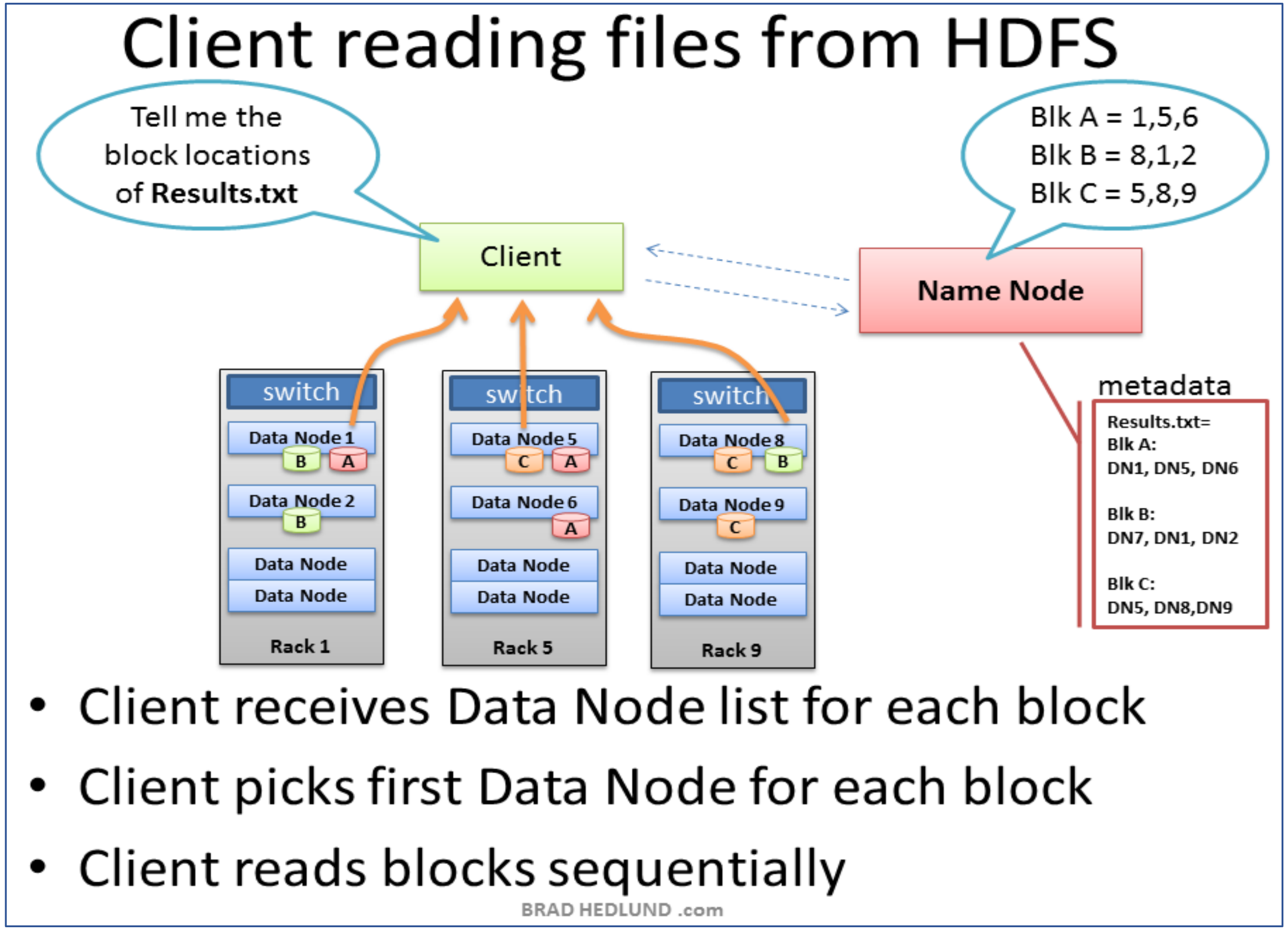
HDFS –Writing a file



- The write operation is outlined in the figure
- Can you outline the steps in the read operation?

Image source: Hadoop Definitive Guide, Tom White, O'Reilly

HDFS: File reading



Client reading files from HDFS

- Client receives Data Node list for each block
- Client picks first Data Node for each block
- Client reads blocks sequentially

BRAD HEDLUND .com

http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/

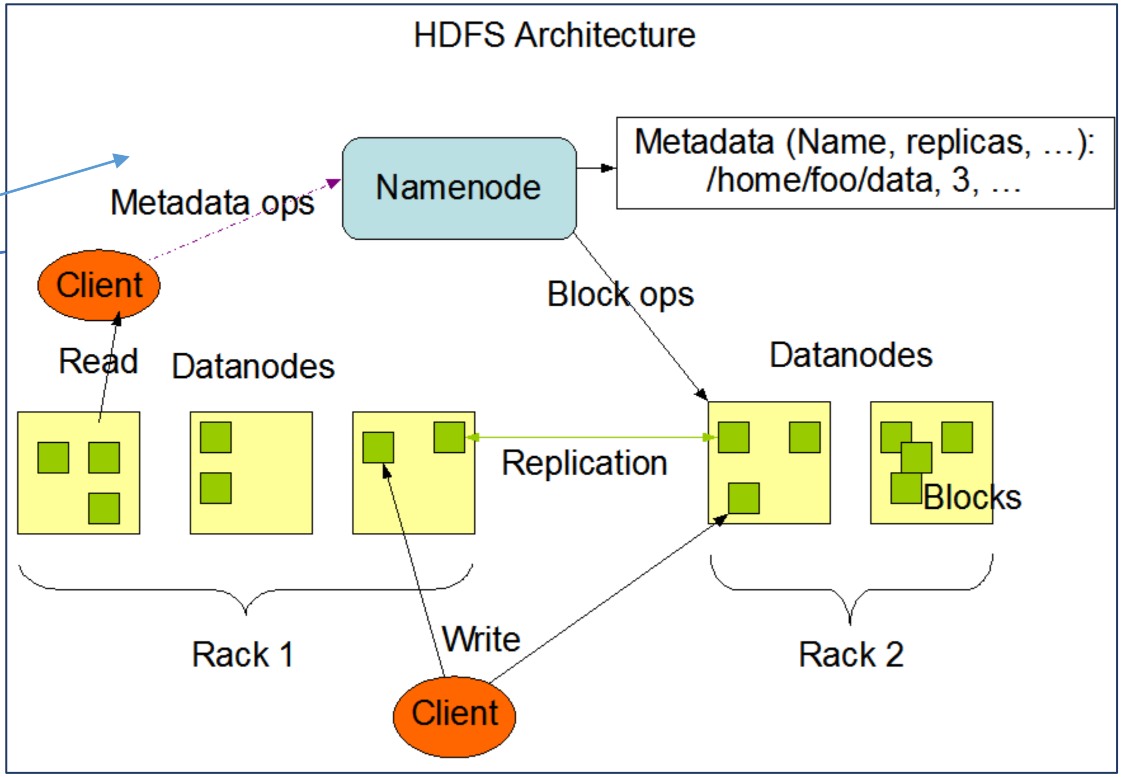
Big Data: HDFS Internals

HDFS –Hadoop distributed File system

HDFS Architecture

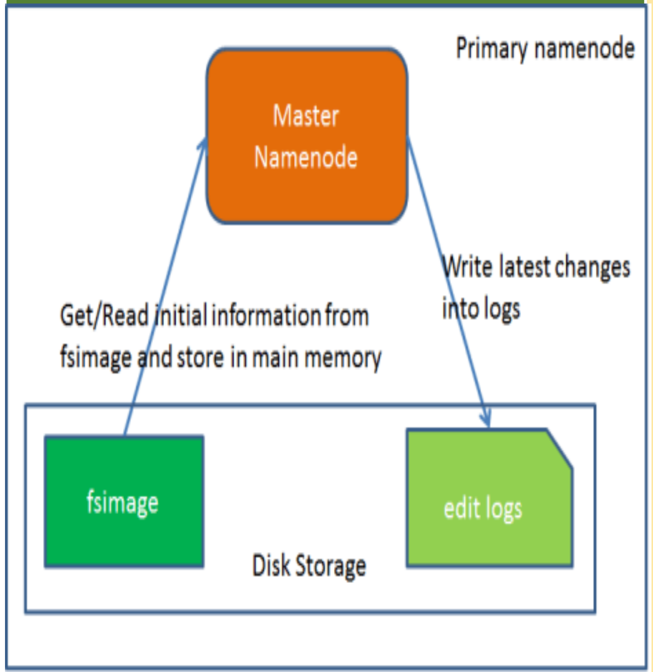
File Metadata (master)

File Data (workers)



https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

Namenode features


Primary namenode
Master Namenode
Write latest changes into logs
Get/Read initial information from fsimage and store in main memory
fsimage
Disk Storage
edit logs

- Manages file system namespace
- Regulate access to files by client
- Opening closing and renaming files
- Manages block creation, deletion and replication
- Determines mapping of blocks to data nodes
- Handles block failure
- Transaction Log
- Contains the metadata in memory

Image source: https://hrouhani.org/hdfs-file-system/

Namenode organization

FSImage

- Serialized form of filesystem tree
- Not updated on every write
  - To avoid recopy of data
- Stores
  - Filename
  - access time
  - #blocks
  - blocks consisting the file

Edit Log

- Every write first writes to edit log
- Flushed and synced after every transaction
- Append only operation

Since no modify operation is done on either file, it can be done really fast.

Namenode memory requirements

- Stored in memory
- What takes up space?
  - #blocks/file
  - Filename length
  - #directories
- Limited amount of memory
- Rule of thumb –1000MB per million blocks
- Solution:
  - Limit the responsibility of each node

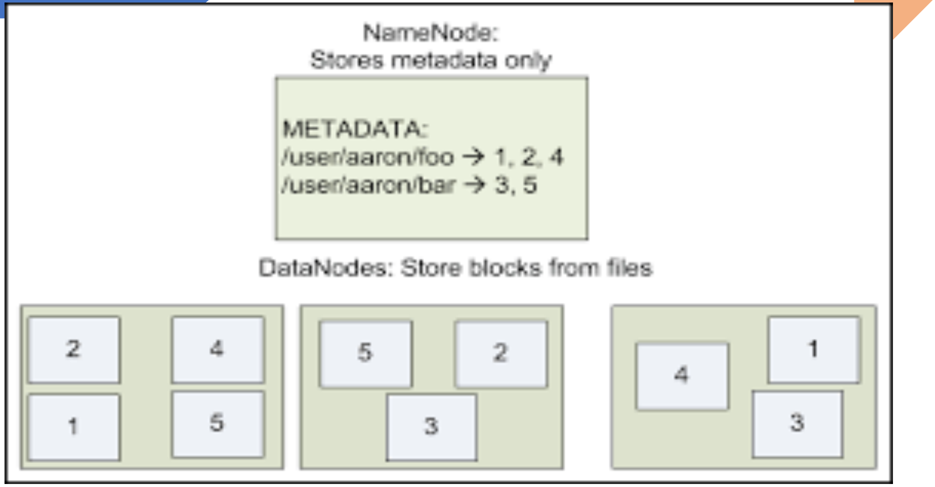
- Example calculation
  - 200 node cluster
  - 24TB/node
  - 128MB block size
  - Replication factor = 3
- How much space is required?
  - #blocks = 200*24*2^20/(128*3)
  - ~<u>12Million blocks</u>
  - ~12,000 MB memory.

HDFS –Data Nodes

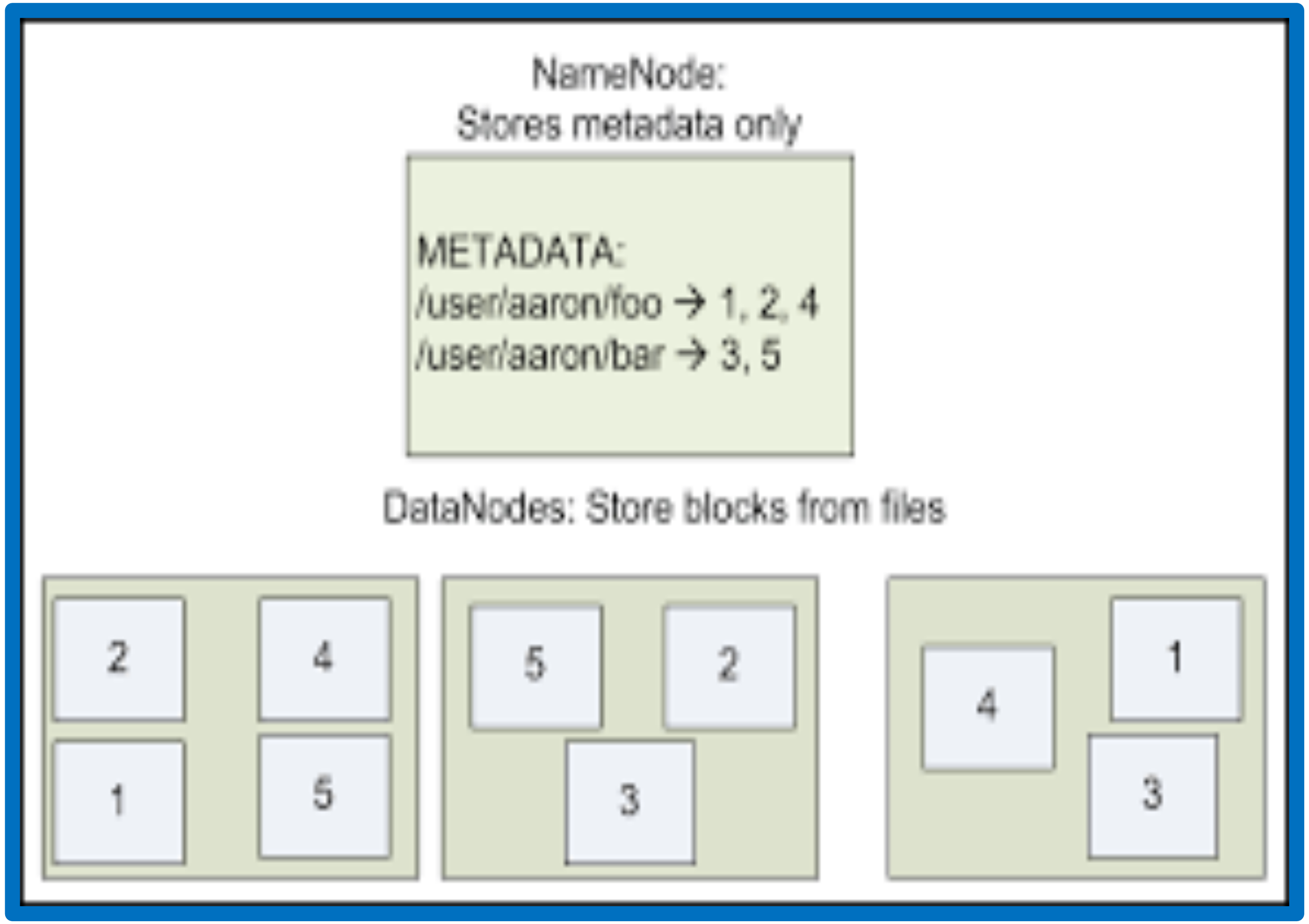
Have lots of disk storage and moderate amounts of processing capabilities and DRAM

Responsible to store the data and process the computation tasks

Periodically sends a report to the name node.



HDFS –Data Nodes



HDFS –Hadoop distributed File system

Data Blocks

Each file split into datablocks –128MB for HDFS v2, 64MB for HDFS v1

Each block is stored on one or more nodes

Each copy of the block is called replica

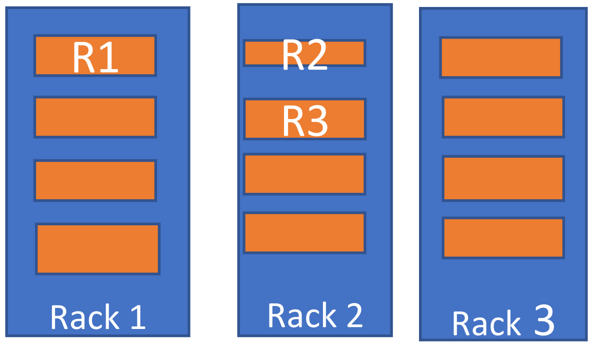
Block placement policy

First replica is placed on the local node

Second replica is placed in a different rack

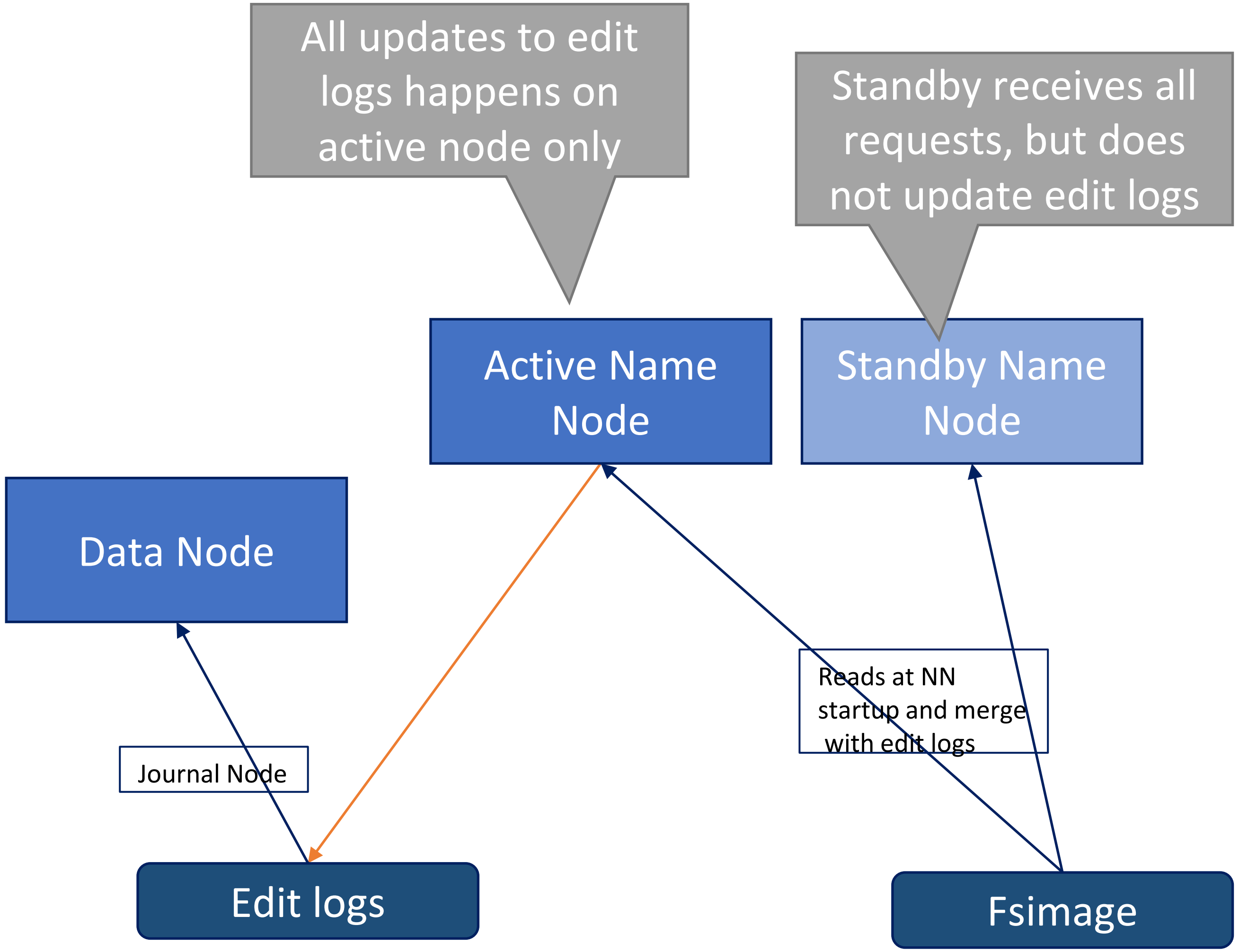
Third replica is placed in the same rack as the second replica

Block placement policy

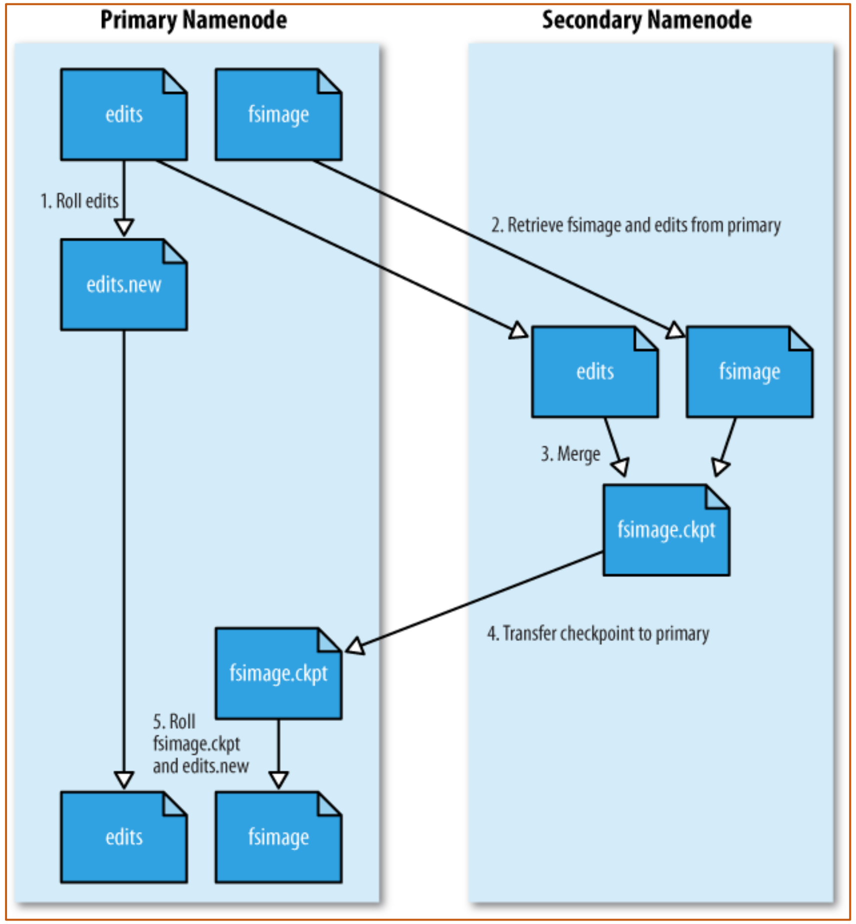


Big Data: HDFS Fault Tolerance and Replication

Namenode fault tolerance


All updates to edit logs happens on active node only
Standby receives all requests, but does not update edit logs
Active Name Node
Standby Name Node
Data Node
Journal Node
Reads at NN startup and merge with edit logs
Edit logs
Fsimage

HDFS : Secondary Namenodes - Operation



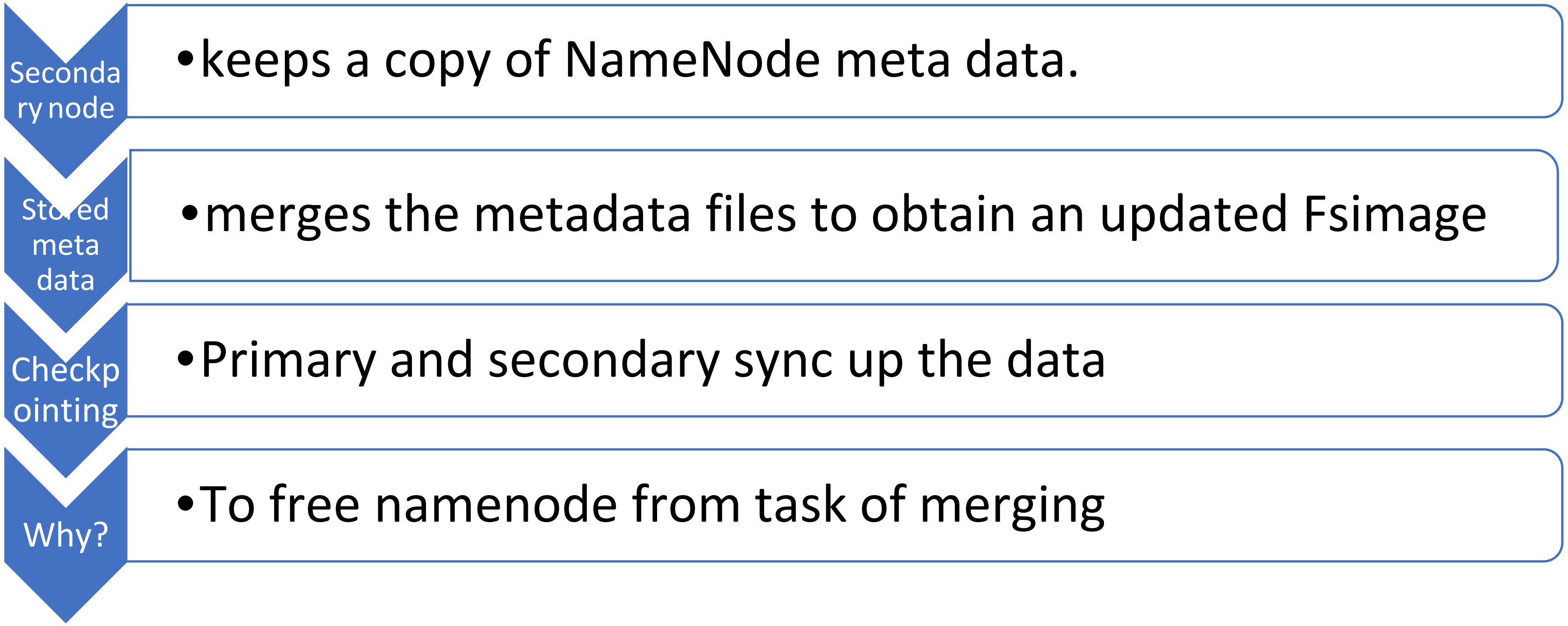
Used to combine fsimage with edits

Offload the operation to secondary

Primary can focus on serving request

On completion of merging, new fsimage will be used.

Secondary Namenodes



Secondary node
- keeps a copy of NameNode meta data.

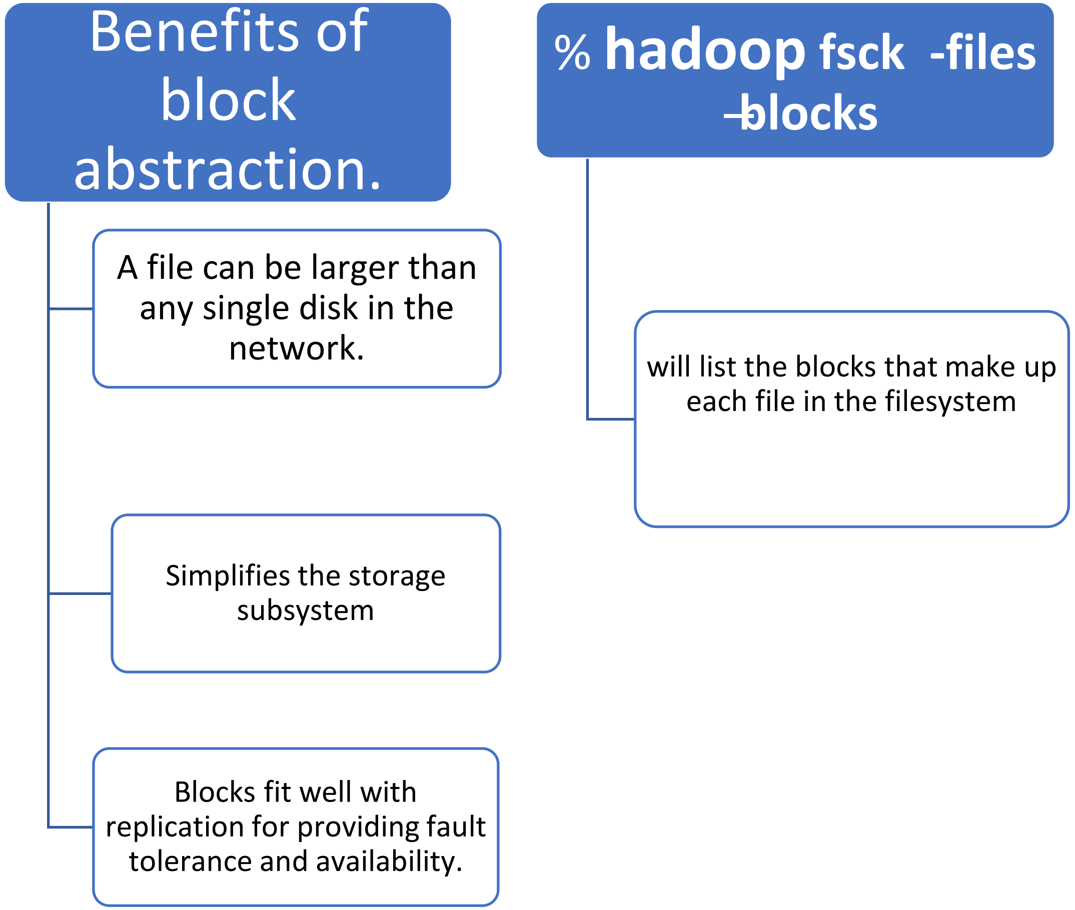
Stored meta data
- merges the metadata files to obtain an updated Fsimage

Checkpointing
- Primary and secondary sync up the data

Why?
- To free namenode from task of merging

Big Data: HDFS Blocks

HDFS blocks - why?



Benefits of block abstraction.

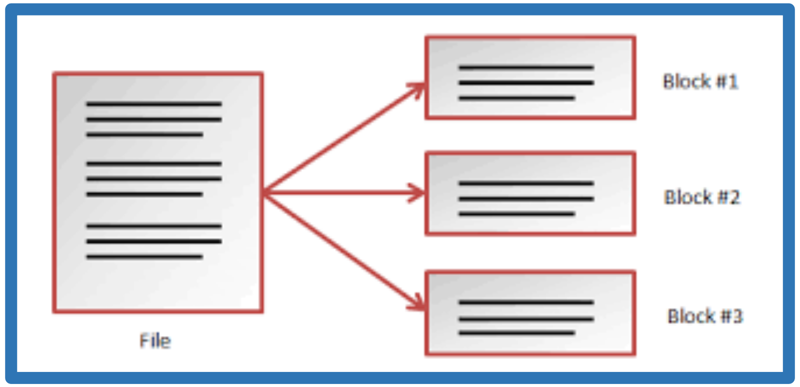
A file can be larger than any single disk in the network.

Simplifies the storage subsystem

Blocks fit well with replication for providing fault tolerance and availability.

% hadoop fsck -files -blocks

will list the blocks that make up each file in the filesystem



HDFS block size –why 128MB?

- Time to read data from disk
  - = seek time + rotational latency + transfer time
  - Seek time –time to position head onto track (variable) - mechanical
  - Rotational latency –time to spin disk to get sector under head (fixed)
  - Transfer time –time to read data (fixed)
- Improving performance of read reduce seek times.
- Large block size transfer time >> seek time

Example

If the seek time is around 10 ms, and the transfer rate is 100 MB/s, then to make the seek time 1% of the transfer time, we need to make the block size around 100 MB.

Hadoop v1 default –64MB

Hadoop v2 default –128MB

Review: why has this increased?

THANK YOU

K V Subramaniam

Department of Computer Science and Engineering

subramaniamkv@pes.edu

+91 80 6666 3333 Extn 877

Content Creators

Prof Resma K S
Prof Prafullata K A
Prof Usha Devi
Prof Rachana