Big Data

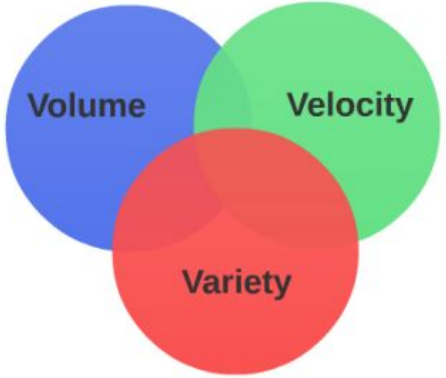
WHAT IS BIG DATA?

WHAT IS BIG DATA?

Many Terabytes, Petabytes, Exabytes...

Name	Abbr.	Size
Kilo	K	1,024
Mega	M	1,048,576
Giga	G	1,073,741,824
Tera	T	1,099,511,627,776
Peta	P	1,125,899,906,842,624
Exa	E	1,152,921,504,606,846,976
Zetta	Z	1,180,591,620,717,411,303,424
Yotta	Y	1,208,925,819,614,629,174,706,176

3Vs - Volume Velocity Variety



IS THERE REALLY A USE CASE?

Science

- Large Hadron Collider - 1 Petabyte every second
- NASA - 1.73 Gigabyte every hour

Government

- NSA - Utah Data Center - Yottabyte Capacity
- Big Data Research and Development Initiative
- Barack Obama's successful 2012 re-election campaign

Private

- eBay - 40PB Hadoop cluster for search, consumer recommendations, and merchandising
- Facebook - 30 PB Hadoop cluster. 50 billion photos. 130TB of logs every day.

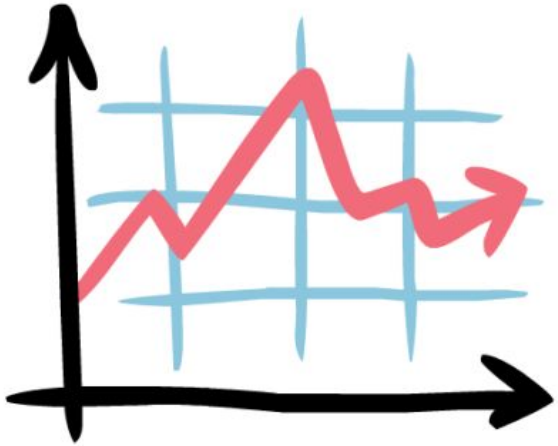
BIG DATA - CHALLENGES

Storage

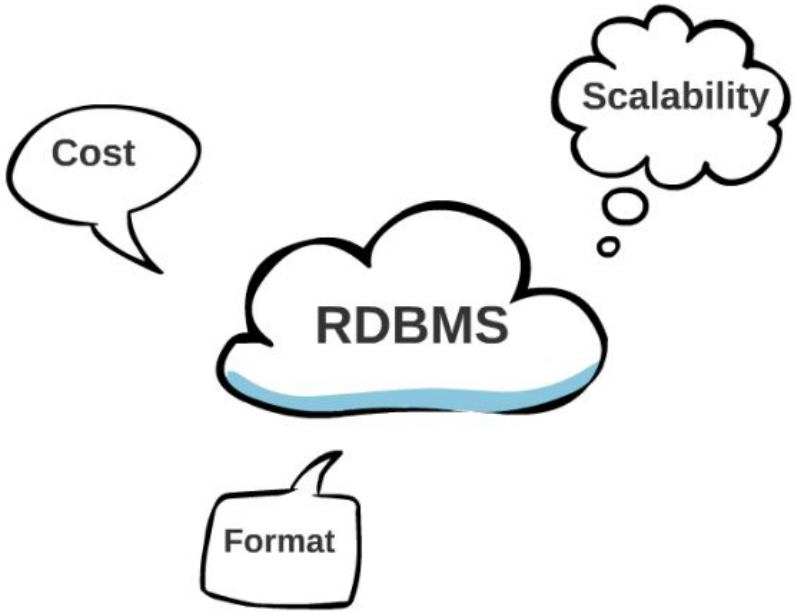
Computational Efficiency

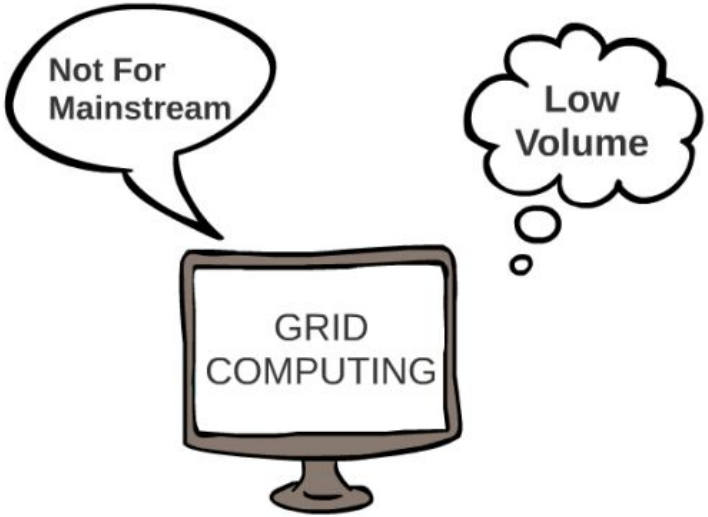
Data Loss

Cost



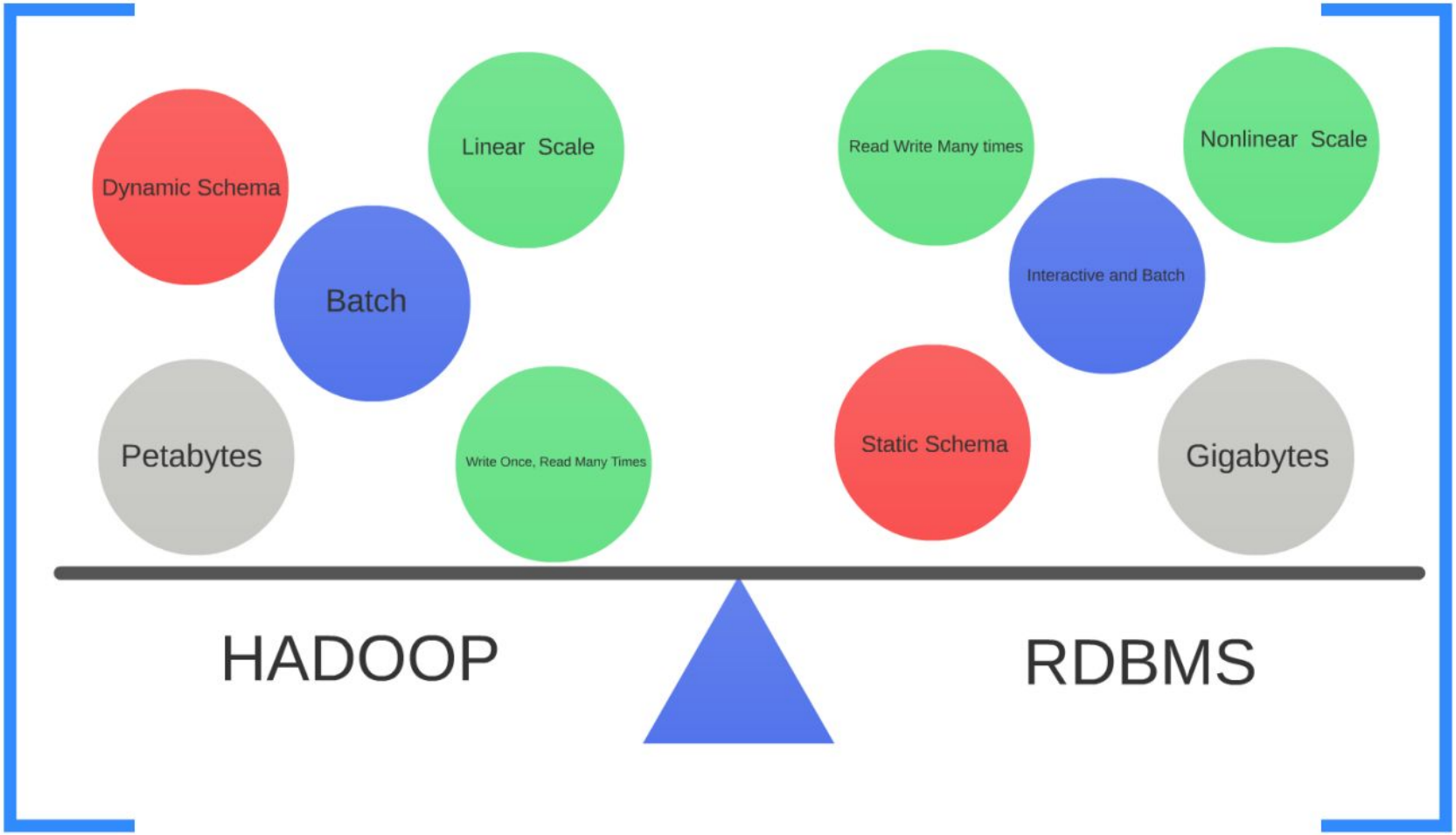
TRADITIONAL SOLUTIONS





HADOOP - A GOOD SOLUTION

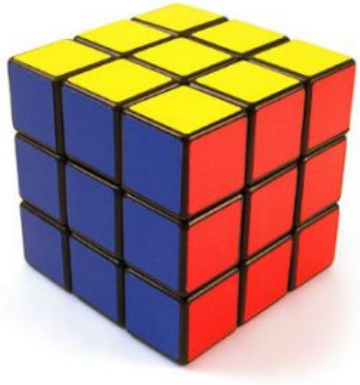
- Support Huge Volume
- Storage Efficiency
- Good Data Recovery Solution
- Horizontal Scaling
- Cost Effective
- Easy For Programmers & Non Programmers


Dynamic Schema
Linear Scale
Batch
Petabytes
Write Once, Read Many Times
Read Write Many times
Nonlinear Scale
Interactive and Batch
Static Schema
Gigabytes
HADOOP
RDBMS

UNDERSTANDING BIG DATA PROBLEM

PROBLEM

ANALYZE



SOLUTION

SAMPLE BIG DATA PROBLEM

- Stocks Dataset - Day by day stock information for several symbols for several years
- Size - 1 TB
- Problem - Find out Maximum closing price for each stock symbol

```
ABCSE,B7J,2008-10-28,6.48,6.74,6.22,6.72,44300,5.79
ABCSE,B7J,2008-10-27,6.21,6.78,6.21,6.40,55200,5.51
ABCSE,B7J,2008-10-24,6.39,6.66,6.21,6.40,67400,5.51
ABCSE,B7J,2008-10-23,6.95,6.95,6.50,6.59,59400,5.68
ABCSE,B7J,2008-10-22,6.92,7.17,6.80,6.80,55300,5.86
ABCSE,B7J,2008-10-21,7.20,7.30,7.10,7.10,54400,6.11
ABCSE,B7J,2008-10-20,6.94,7.31,6.94,7.12,45700,6.13
ABCSE,B7J,2008-10-17,6.43,6.93,6.42,6.90,57700,5.94
ABCSE,B7J,2008-10-16,6.61,6.69,6.21,6.53,83200,5.62
ABCSE,B7J,2008-10-15,6.84,6.90,6.36,6.36,78900,5.48
ABCSE,B7J,2008-10-14,7.15,7.32,6.93,6.96,74700,5.99
ABCSE,B7J,2008-10-13,6.00,6.57,6.00,6.57,75700,5.66
ABCSE,B7J,2008-10-10,5.05,5.72,4.79,5.72,158400,4.93
ABCSE,B7J,2008-10-09,6.30,6.41,6.00,6.02,140500,5.18
ABCSE,B7J,2008-10-08,5.60,6.47,5.60,6.28,292000,5.41
ABCSE,B7J,2008-10-07,7.59,7.59,6.66,6.69,89900,5.76
ABCSE,B7J,2008-10-06,7.83,7.90,7.00,7.40,159600,6.37
```

EXECUTION TIME

Data access rate

+

Program computation time (~60 mins)

+

Network Bandwidth.. etc..

> 3 hrs


Average Data Access Rate - 122 MB/sec
1 TB file will take 2 1/2 hours to read from disk
1 Sec - 122 MB
x Secs - 1048576 MB (1 TB)
x = 1048576 / 122 = 8595 secs
2 hr 22 mins

EXECUTION TIME

Data access rate

+

Program computation time (~60 mins)

+

Network Bandwidth.. etc..

> 3 hrs

HOW ABOUT THIS?

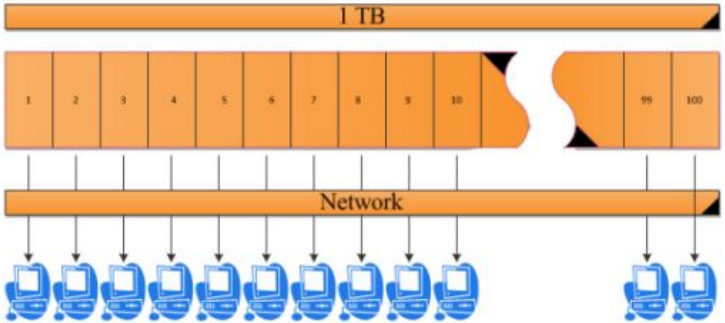
Split 1 TB file in to 100 equal sized blocks and read them parallely

Time to read = 150 mins /100

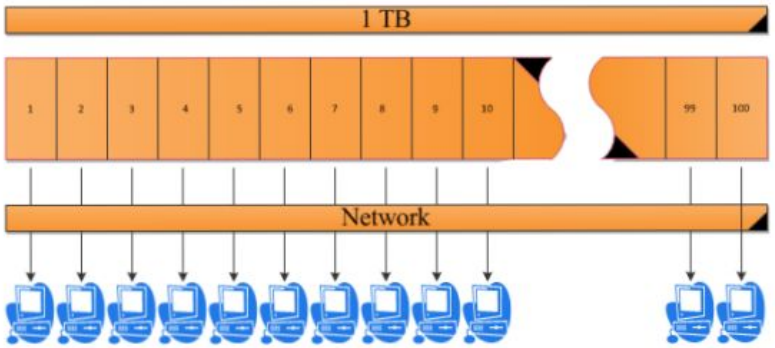
< 2 minutes

Computation Time = 60 mins /100

< 1 minute

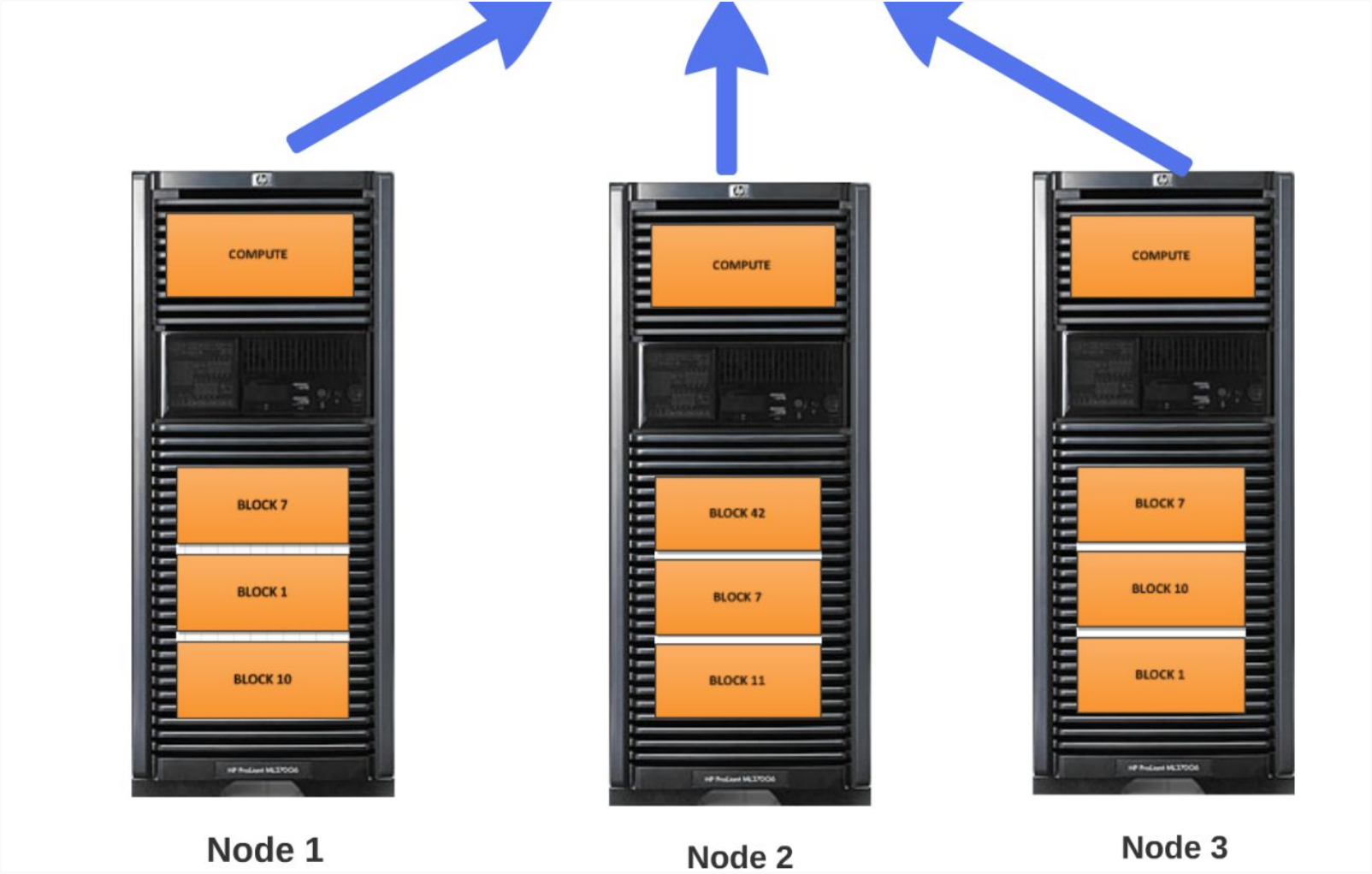


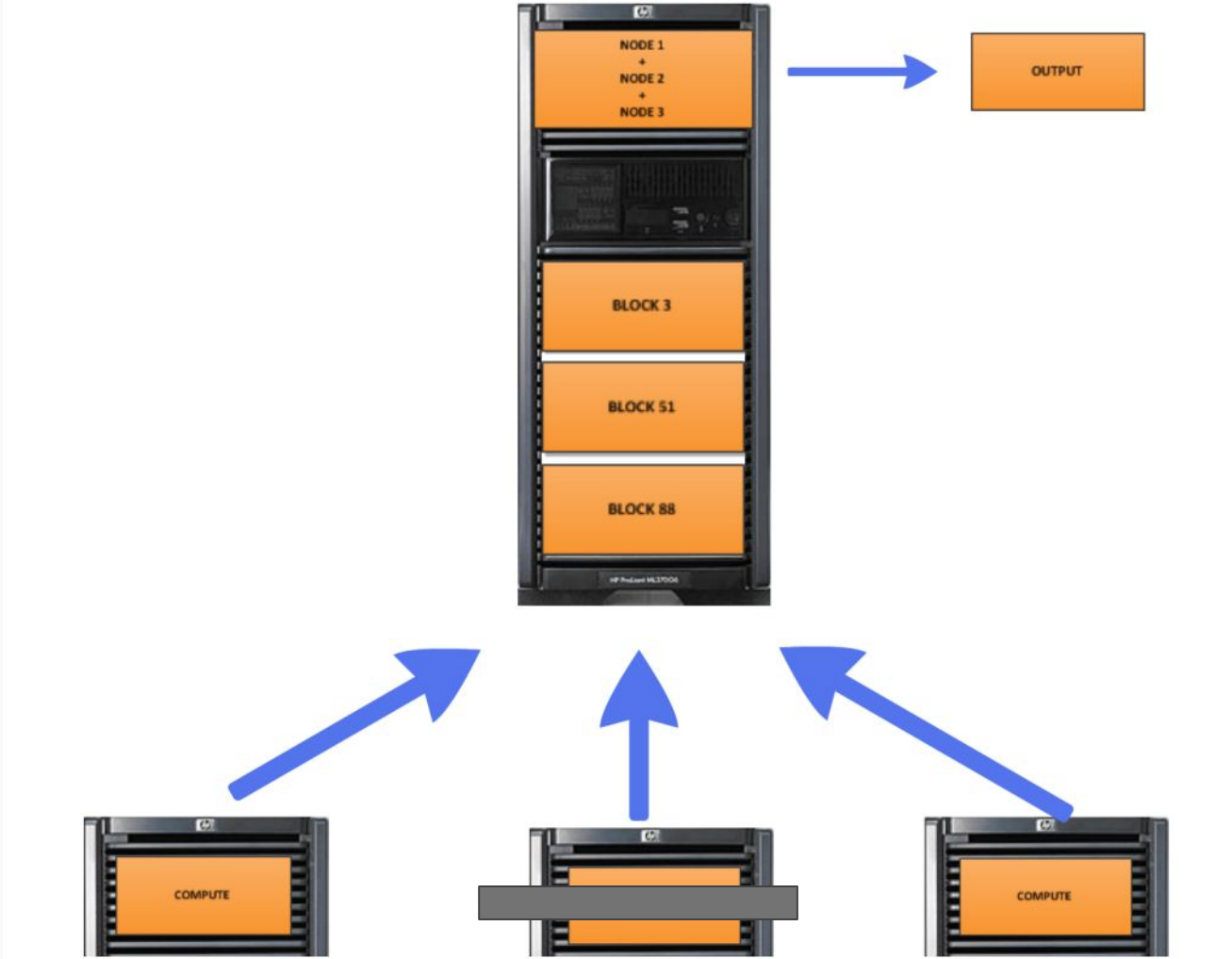
STORAGE CLOSER TO COMPUTATION

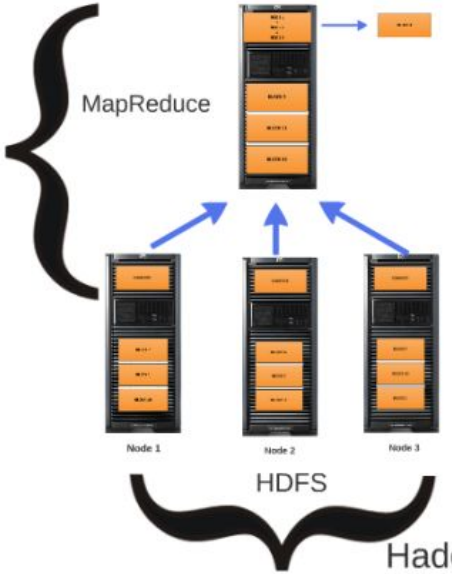


AGGREGATE COMPUTATION

REPLICATION


COMPUTE
BLOCK 7
BLOCK 1
BLOCK 10
COMPUTE
BLOCK 42
BLOCK 7
BLOCK 11
COMPUTE
BLOCK 7
BLOCK 10
BLOCK 1
Node 1
Node 2
Node 3


NODE 1
+
NODE 2
+
NODE 3
OUTPUT
BLOCK 3
BLOCK 51
BLOCK 88
COMPUTE
COMPUTE



HDFS - Reliable Shared Storage

+

MapReduce - Distributed Computation

=

hadoop

Hadoop is a framework for distributed processing of large data sets across clusters of commodity computers

Doug Cutting adds DFS & MapReduce support to Nutch

YAHOO!

Fastest sort of a TB, 3.5mins over 910 nodes

Fastest sort of a TB, 62secs over 1,460 nodes
Sorted a PB in 16.25hours over 3,658 nodes

Doug Cutting & Mike Cafarella started working on Nutch

NY Times converts 4TB of image archives over 100 EC2s

2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009

Google publishes GFS & MapReduce papers

Google

Yahoo! hires Cutting, Hadoop spins out of Nutch

cloudera
Founded

Doug Cutting joins Cloudera

Facebooks launches Hive: SQL Support for Hadoop

facebook

Hadoop Summit 2009, 750 attendees

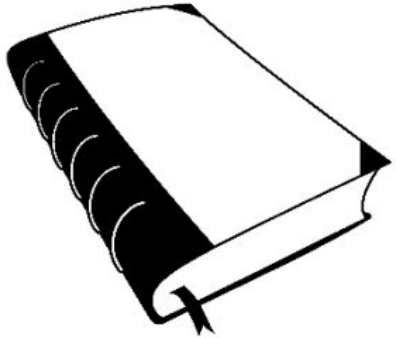
Michael j. cafarella

Doug cutting

HADOOP DISTRIBUTED FILE SYSTEM

PILE OF PAPERS VS. BOOK

VS



Go to Chapter 34 - Act 2

Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins.

FUNCTIONS OF FILE SYSTEM

- Control how data is stored and retrieved
- Metadata about the files and folders
- Permissions and security
- Manage storage space efficiently

DIFFERENT FILE SYSTEMS

Microsoft

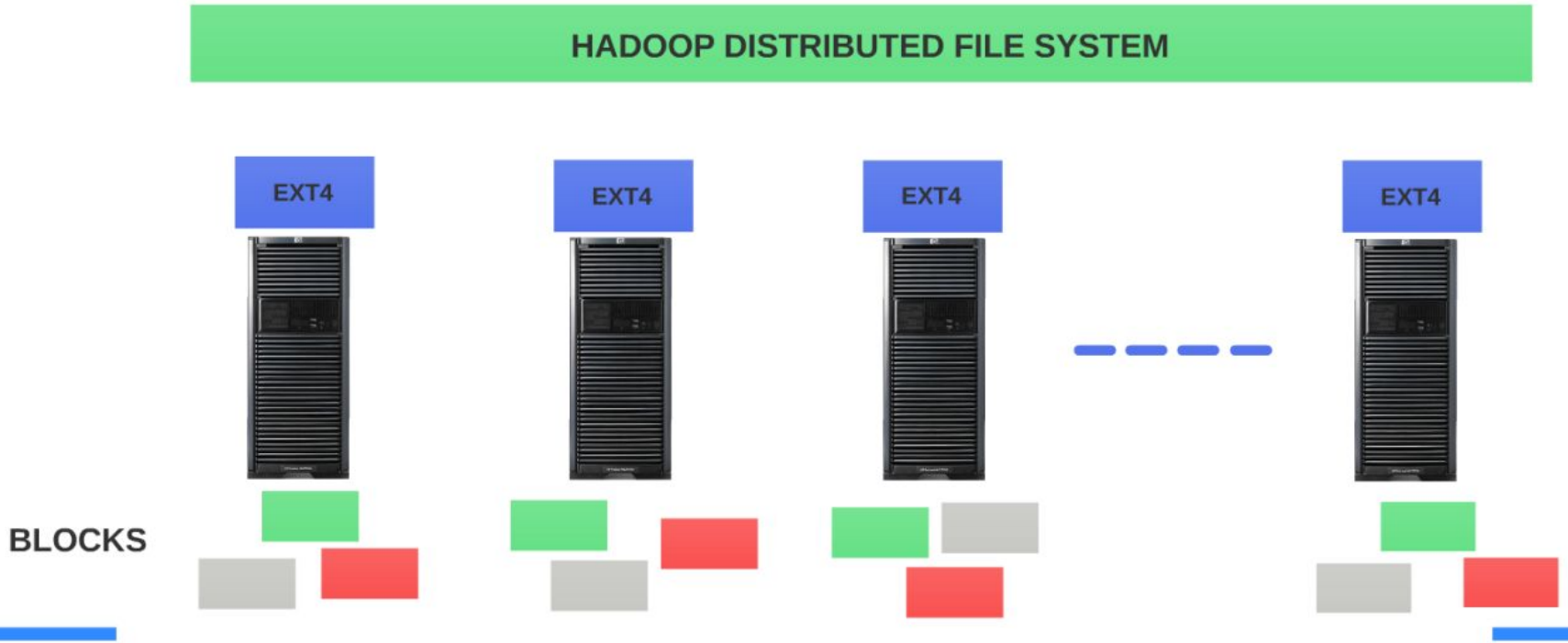
FAT32 - 4 GB File limit 32 GB Volume limit
NTFS - 16 EB File limit 16 EB Volume limit

HFS - 2 GB File limit 2 TB Volume limit
HFS+ - 8 EB File limit 8 EB Volume limit

ext3 - 2 TB File limit 32 TB Volume limit
ext4 - 16 TB File limit 1 EB Volume limit
XFS - 8 EB File limit 8 EB Volume limit

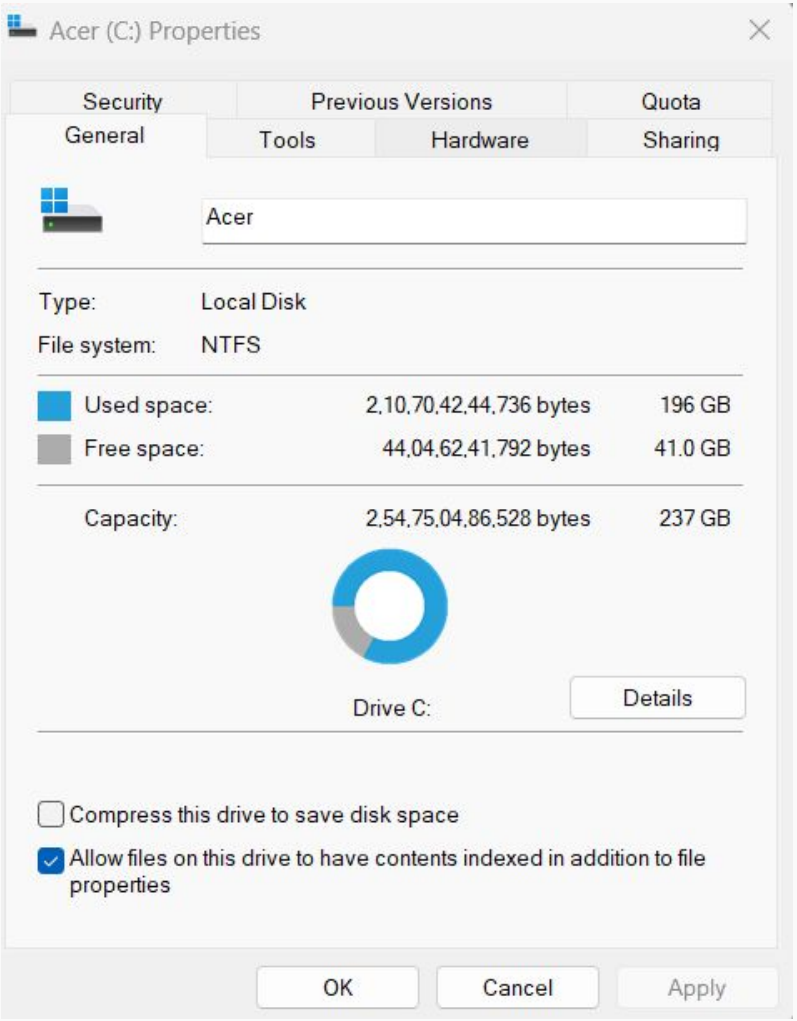
Why another file system ?

LOCAL FILE SYSTEM vs. HDFS



BENEFITS OF HDFS

- Support distributed processing
  - Blocks (not as whole files)
- Handle failures
  - Replicate blocks
- Scalability
  - Able to support future expansion
- Cost effective
  - Commodity hardware


Acer (C:) Properties
Security
Previous Versions
Quota
General
Tools
Hardware
Sharing
Acer
Type: Local Disk
File system: NTFS
Used space: 2,10,70,42,44,736 bytes 196 GB
Free space: 44,04,62,41,792 bytes 41.0 GB
Capacity: 2,54,75,04,86,528 bytes 237 GB
Drive C:
Details
Compress this drive to save disk space
Allow files on this drive to have contents indexed in addition to file properties
OK
Cancel
Apply

Why do we need a file system in the first place, you may ask?

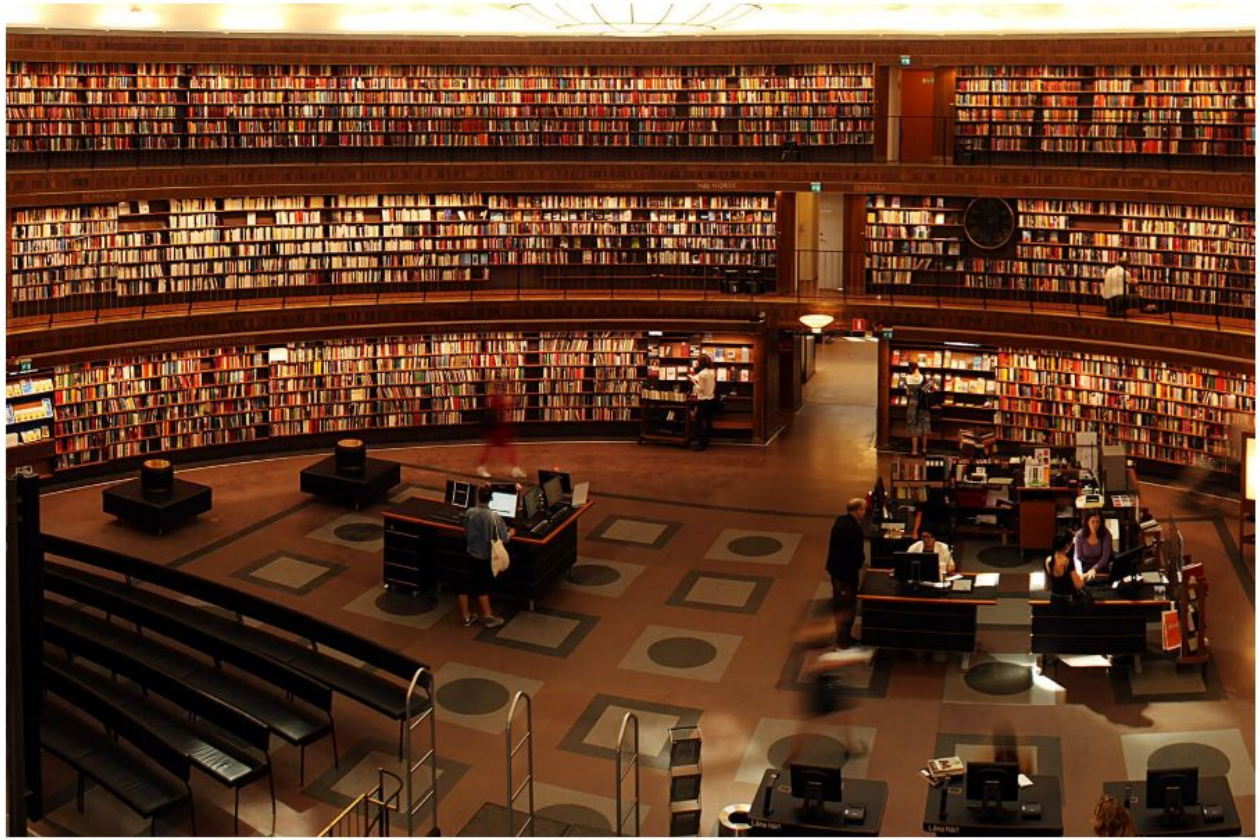
Well, without a file system, the storage device would contain a big chunk of data stored back to back, and the operating system wouldn't be able to tell them apart.

The term file system takes its name from the old paper-based data management systems, where we kept documents as files and put them into directories.

Imagine a room with piles of papers scattered all over the place.

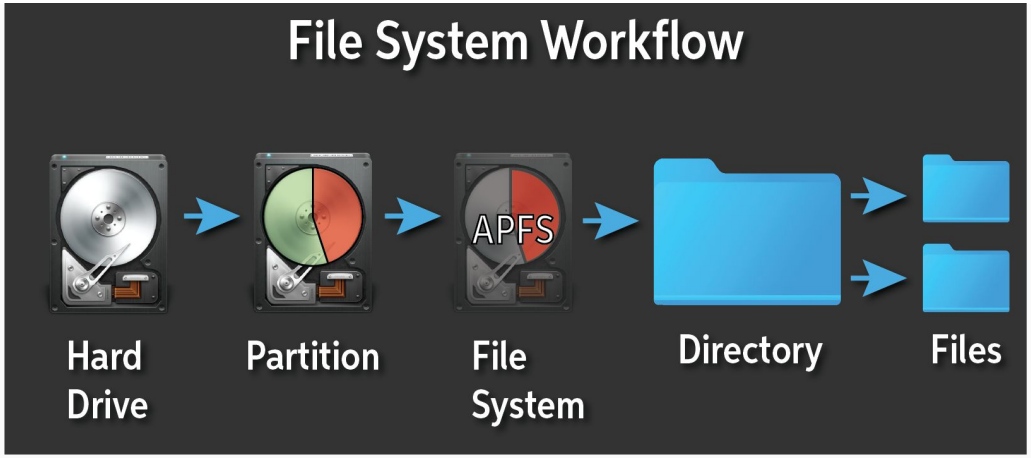
A storage device without a file system would be in the same situation - and it would be a useless electronic device.

However, a file system changes everything:



A file system isn't just a bookkeeping feature, though.

Space management, metadata, data encryption, file access control, and data integrity are the responsibilities of the file system too.



Everything begins with partitioning

Storage devices must be **partitioned** and **formatted** before the first use.

But what is partitioning?

Partitioning is splitting a storage device into several *logical regions*, so they can be managed separately as if they are separate storage devices.

We usually do partitioning by a disk management tool provided by operating systems, or as a command-line tool provided by the system's firmware.

A storage device should have at least one partition or more if needed.

Why should we split the storage devices into multiple partitions anyways?

The reason is that we don't want to manage the whole storage space as a single unit and for a single purpose.It's just like how we partition our workspace, to separate (and isolate) meeting rooms, conference rooms, and various teams.

What do you mean by firmware?

Firmware Definition

Firmware provides instructions to help hardware start up, communicate with other devices, and perform basic input/output tasks. Software, on the other hand, is installed onto a device and used for interaction, such as browsing the internet, word processing, listening to music, and videoconferencing.

NTFS

4 KB BLOCK SIZE

FILE SIZE	UNUSED SPACE
2 KB	2 KB
8 KB	0 KB
13 KB	3 KB

HDFS

256 MB BLOCK SIZE

FILE SIZE	UNUSED SPACE
1 MB	?

HDFS
READ
WRITE

Namenode

HDFS - Metadata
Block locations

Datanode

Stores actual blocks

Read Operation

Name Node

Give me block locations for MyFirstFileInHDFS.log

BLK_0045732	R8 DN20	R1 DN2	R1 DN10
BLK_9610590	R8 DN20	R3 DN4	R3 DN13
BLK_8851209	R2 DN7	R1 DN2	R1 DN10

Client

Data Nodes

Send me BLK_0045732

Here you go

R8 DN20

Send me BLK_9610590

Here you go

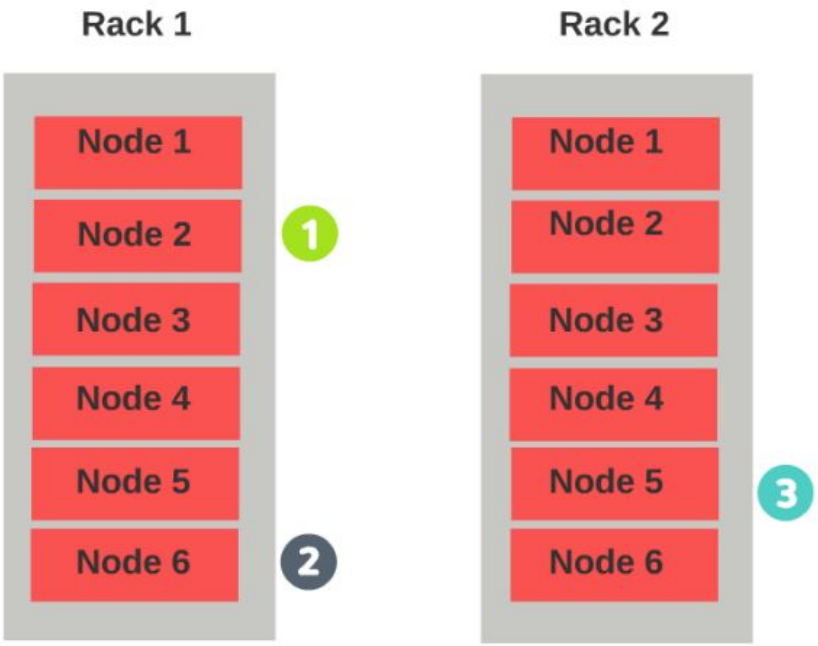
R3 DN4

Send me BLK_8851209

Here you go

R2 DN7

NODE PROXIMITY



Write Operation

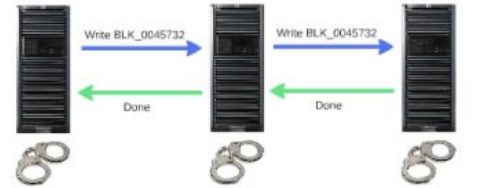
Give me block locations for
MySecondFileInHDFS.log

BLK_0045732	R8 DN20	R1 DN2	R1 DN10
BLK_9610590	R8 DN20	R3 DN4	R3 DN13
BLK_8851209	R2 DN7	R1 DN2	R1 DN10

Done

Write BLK_0045732

Done



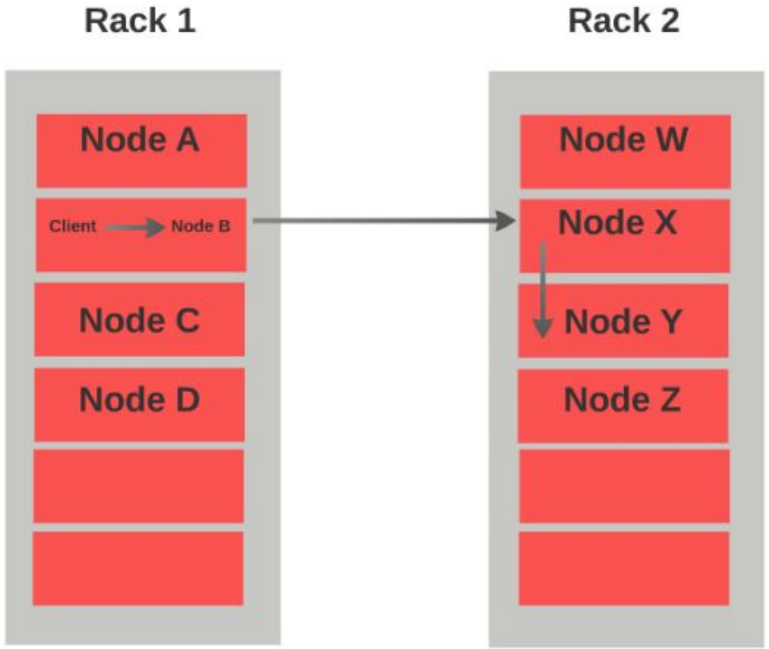
Name Node

Client

Data Nodes Pipeline

Replica Placement

1. Same Node as Client
2. Node in another Rack
3. Node in same Rack as 2



Write Operation - Failure

Name Node

Give me block locations for
MySecondFileInHDFS.log

BLK_0045732	R8 DN20	R1 DN2	R1 DN10
BLK_9610590	R8 DN20	R3 DN4	R3 DN13
BLK_8851209	R2 DN7	R1 DN2	R1 DN10

Client

Write BLK_0045732

(BLK_0045732XXX) Done

Change BLK_0045732 to Write BLK_0045732XXX

R8 DN20

R1 DN2

R1 DN10

Write BLK_0045732XXX

(BLK_0045732XXX) Done

R6 DN12

Data Nodes Pipeline