# Big Data Analytics

Prepared By :- Prof . Shital Pathar

# Question

- What is Data?
- What is information?
- What is knowledge/insights?

- Data □ Information
- Information □ Insights

POPULAR SCIENCE
THE FUTURE NOW
THE CONTROL CENTERS
Using Data to Feed the World, Solve Cold Cases, Battle Malware, Predict Our Fate
OFFICER ALGORITHM
Can a Crime Be Prevented Before It Begins?
NEW WAYS OF SEEING
A Gallery of Extraordinary Infographics
PLUS
Juan Enriquez Reprograms Life
James Gleick Unsplits the Bit
AND
Lawrence Weschler Questions the Cloud
SPECIAL ISSUE
DATA IS POWER
HOW INFORMATION IS DRIVING THE FUTURE

# Data contains value and knowledge

# History

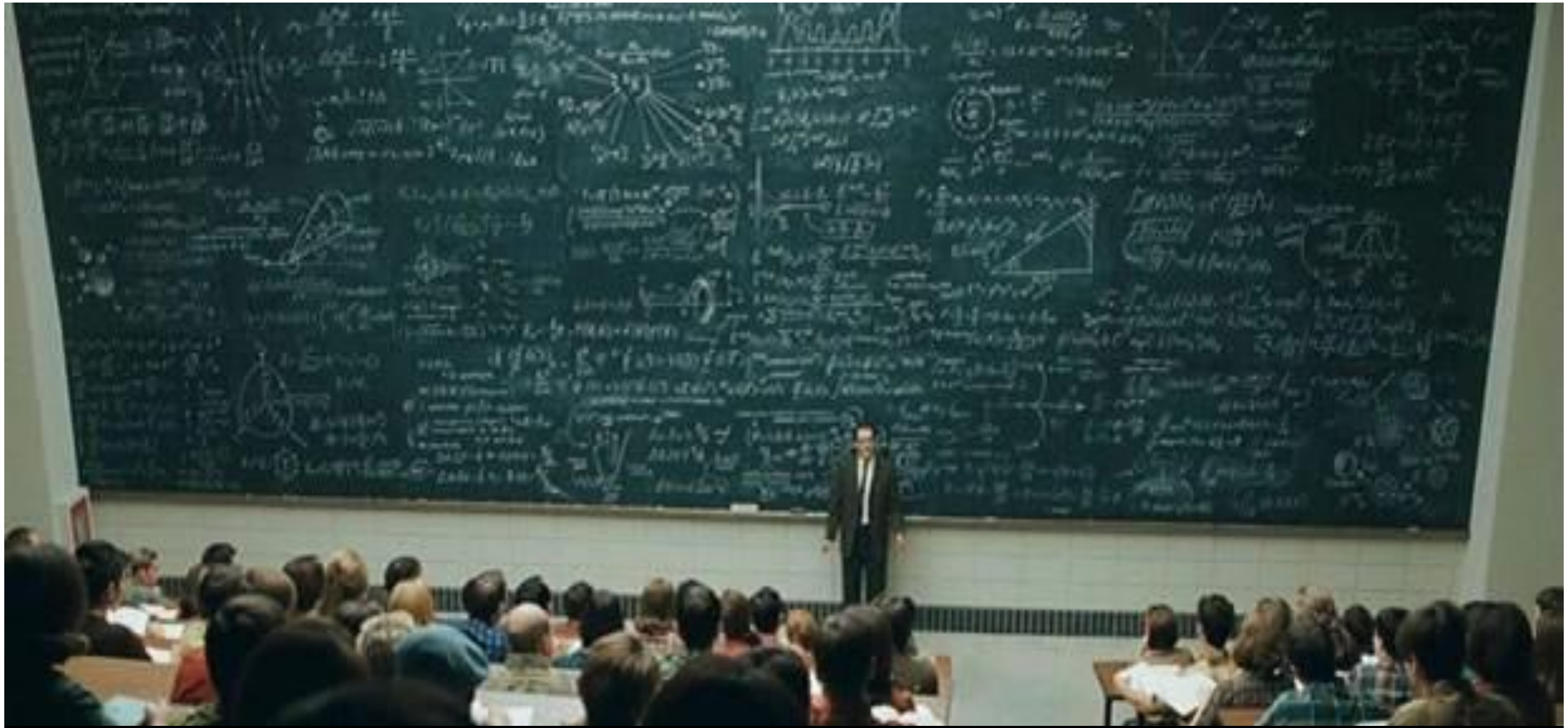
- **Flat file**
    - Text, comma separated file, program file
- **Database Management System**
    - Structure data
- **Relational Database Management System**
    - Codd at IBM in 1970
    - Oracle, MySQL, Microsoft SQL Server, PostgreSQL
- **Data warehousing**
    - Reporting and Data Analysis
    - Historical + Current + Future

# Data Mining

- But to extract the knowledge data needs to be
  - Stored
  - Managed
  - And ANALYZED

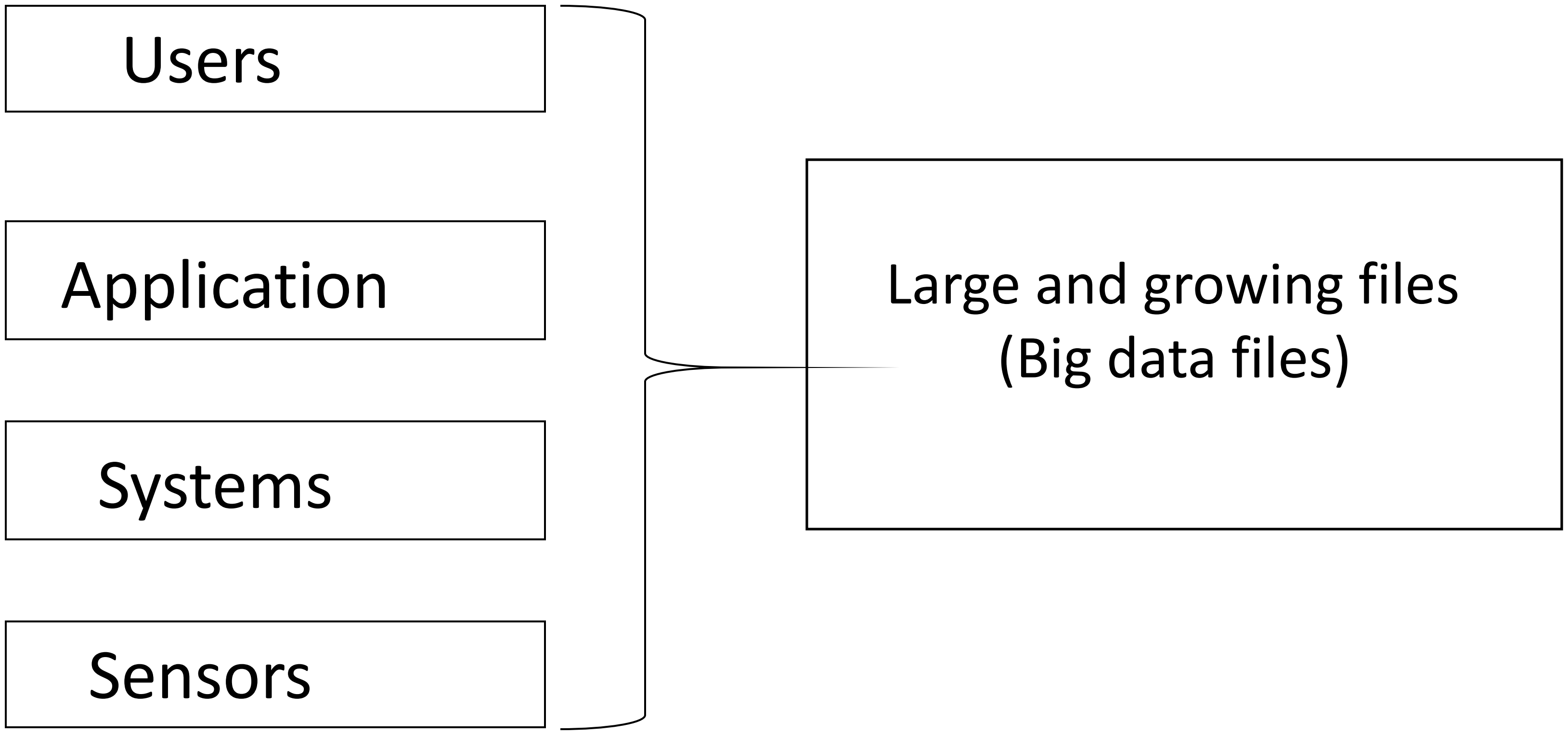
**Data Mining ≈ Big Data ≈ Predictive Analytics ≈ Data Science**

# What is Big Data?



**“A massive volume of both structured and unstructured data that is so large & complex it's difficult  to process with traditional database management tools & software techniques.”**

# Big Data sources


Users

Application

Systems

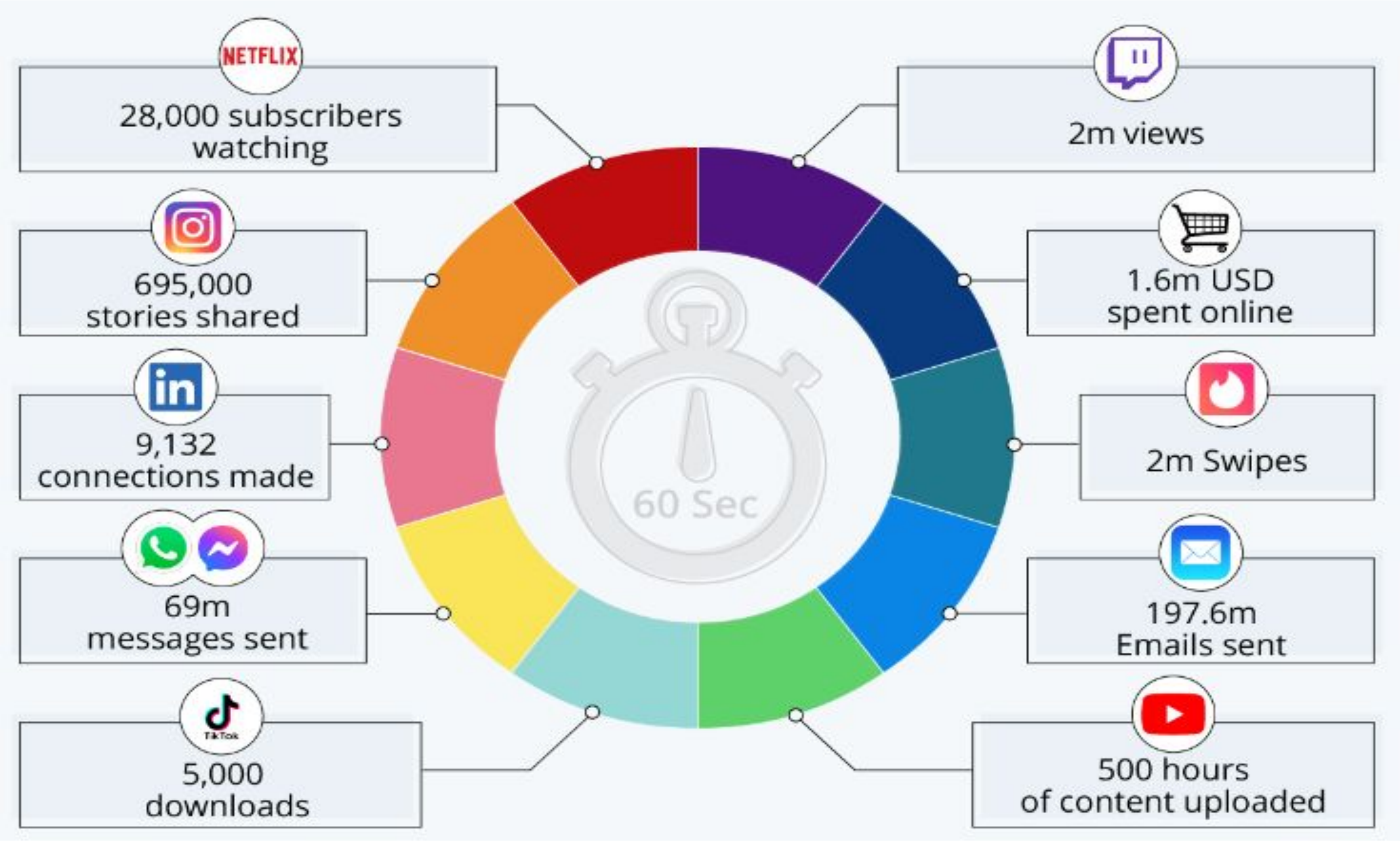
Sensors

Large and growing files
(Big data files)

# Big Data is EveryWhere!

- Lots of data is being collected and warehoused
    - E-commerce, E-shopping
    - Purchases at Department/ Grocery stores
    - Bank/Credit Card transactions
    - Social Network
    - Web Data

# A Minute on the Internet in 2022



# Data Units

| Name | Equal to: | Size in Bytes | | |
|---|---|---|---|---|
| Bit | 1 bit | 1/8 | | |
| Nibble | 4 bits | 1/2 (rare) | | |
| Byte | 8bits | 1 | | |
| Kilobyte | 1,024 bytes | 1,024 | | |
| Megabyte | 1,024 kilobytes | 1,048,576 | | |
| Gigabyte | 1,024 megabytes | 1,073,741,824 | | |
| Terrabyte | 1,024 gigabytes | 1,099,511,627,776 | | |
| Petabyte | 1,024 terrabytes | 1,125,899,906,842,624 | | |
| Exabyte | 1,024 petabytes | 1,152,921,504,606,846,976 | | |
| Zettabyte | 1,024 exabytes | 1,180,591,620,717,411,303,424 | | |
| Yottabyte | 1,024 zettabytes | 1,208,925,819,614,629,174,706,176 | | |

# How much data?

- Google processes 20 PB a day (2008)
- Facebook has 2.5 PB of user data + 15 TB/day (2009)
- eBay has 6.5 PB of user data + 50 TB/day (2009)

# Type of Data

1. Structured data Relational Data (Tables/Transaction)
2. Unstructured data Text Data (Web)
3. Semi-structured Data (XML)

# What to do with these data?

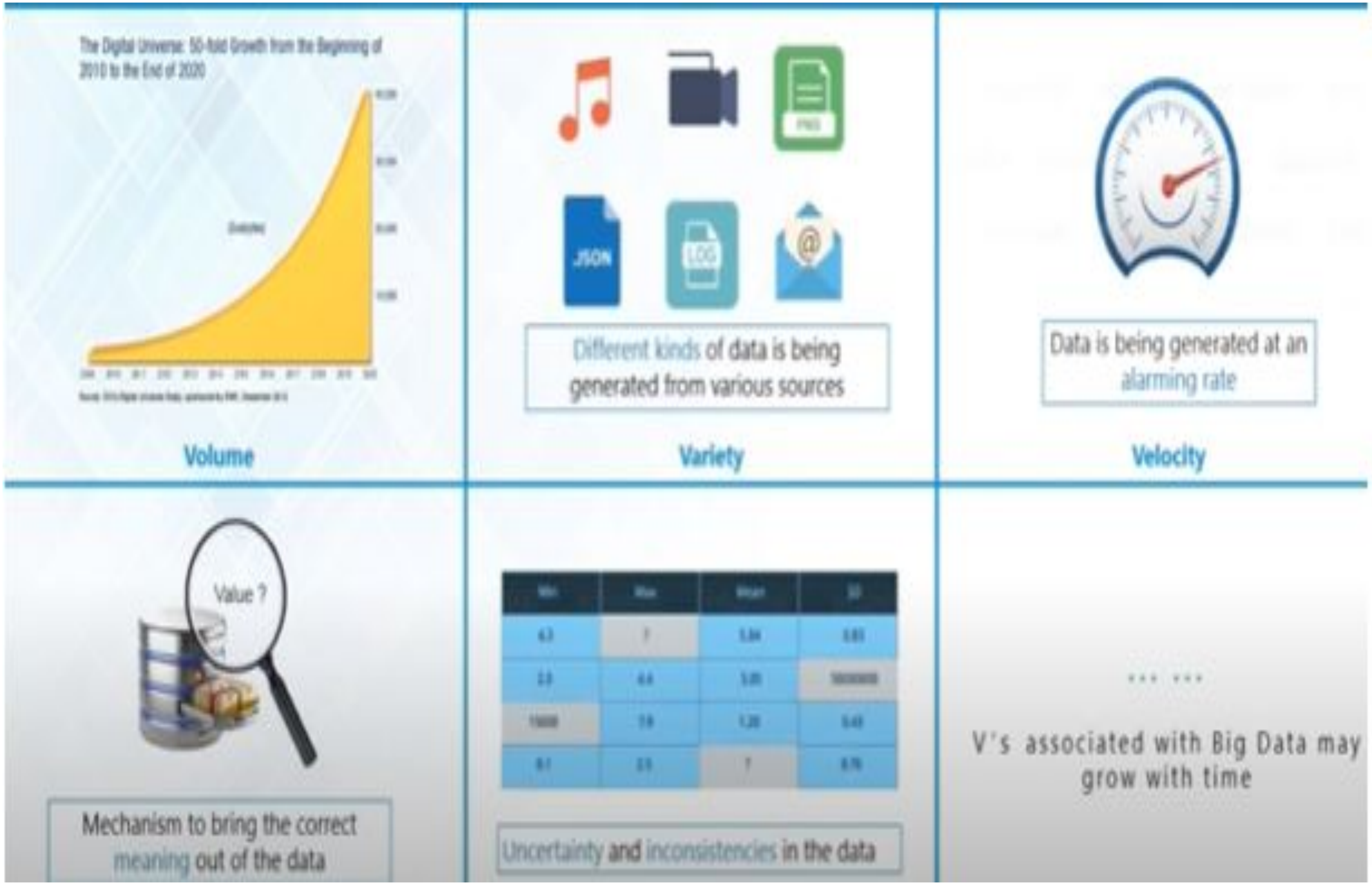
- **Aggregation and Statistics**
  - Data warehouse and OLAP
- **Indexing, Searching, and Querying**
  - Keyword based search
  - Pattern matching
- **Knowledge discovery**
  - Data Mining
  - Statistical Modeling

# Big Data: A definition

- Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools.
- The challenges include capture, cleaning, storage, search, sharing, analysis, and visualization.
- spot business trends, determine quality of research, prevent diseases, link legal citations, combat crime, and determine real-time roadway traffic conditions.

# 5 Vs of Big Data

1. Volume
2. Variety
3. Velocity
4. Veracity
5. Value


The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020
Volume
PNG
JSON
LOG
Variety
Velocity
Value ?

Different kinds of data is being generated from various sources

Data is being generated at an alarming rate

Mechanism to bring the correct meaning out of the data

Uncertainty and inconsistencies in the data

... ...

V's associated with Big Data may grow with time

# Big Data Analytics

- Examining large amount of data
- Appropriate information
- Identification of hidden patterns, unknown correlations
- Competitive advantage
- Better business decisions: strategic and operational
- Effective marketing,  customer satisfaction, increased revenue

# Data analytics vs Data analysis

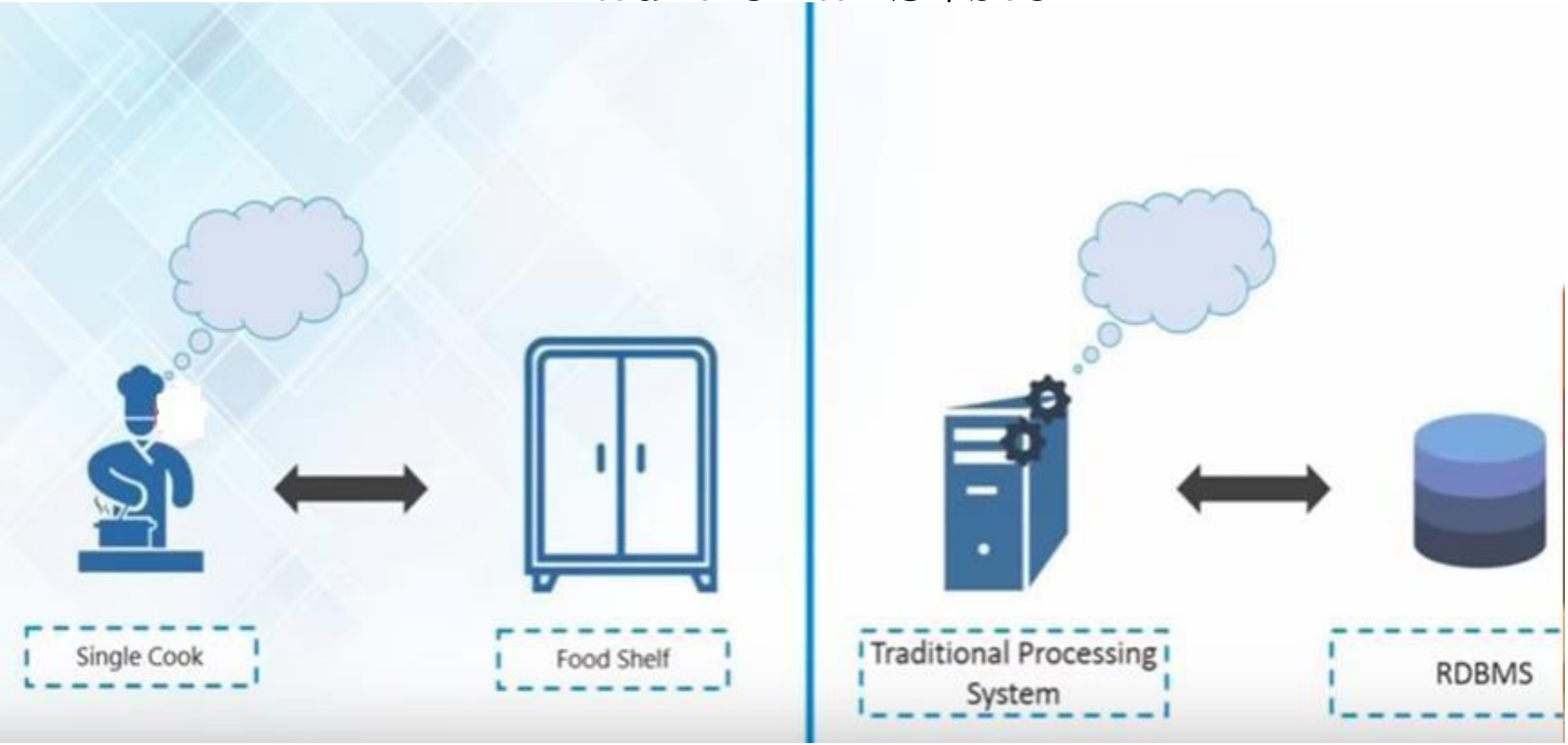
- Most of us are at least somewhat knowledgeable about the stock market. Imagine that you are a newbie and that you want to start your trade with a profit. Now, describe your initial plan of action.

- As a new trader, you've probably researched sharemarket and trend records to get a sense of what's going on in the market. This technique includes **data analysis**.
- As a result of your newfound understanding of the stock pattern, you can now estimate the stock's future market price and purchase some shares. This serves as an example of a **data analytics** process.

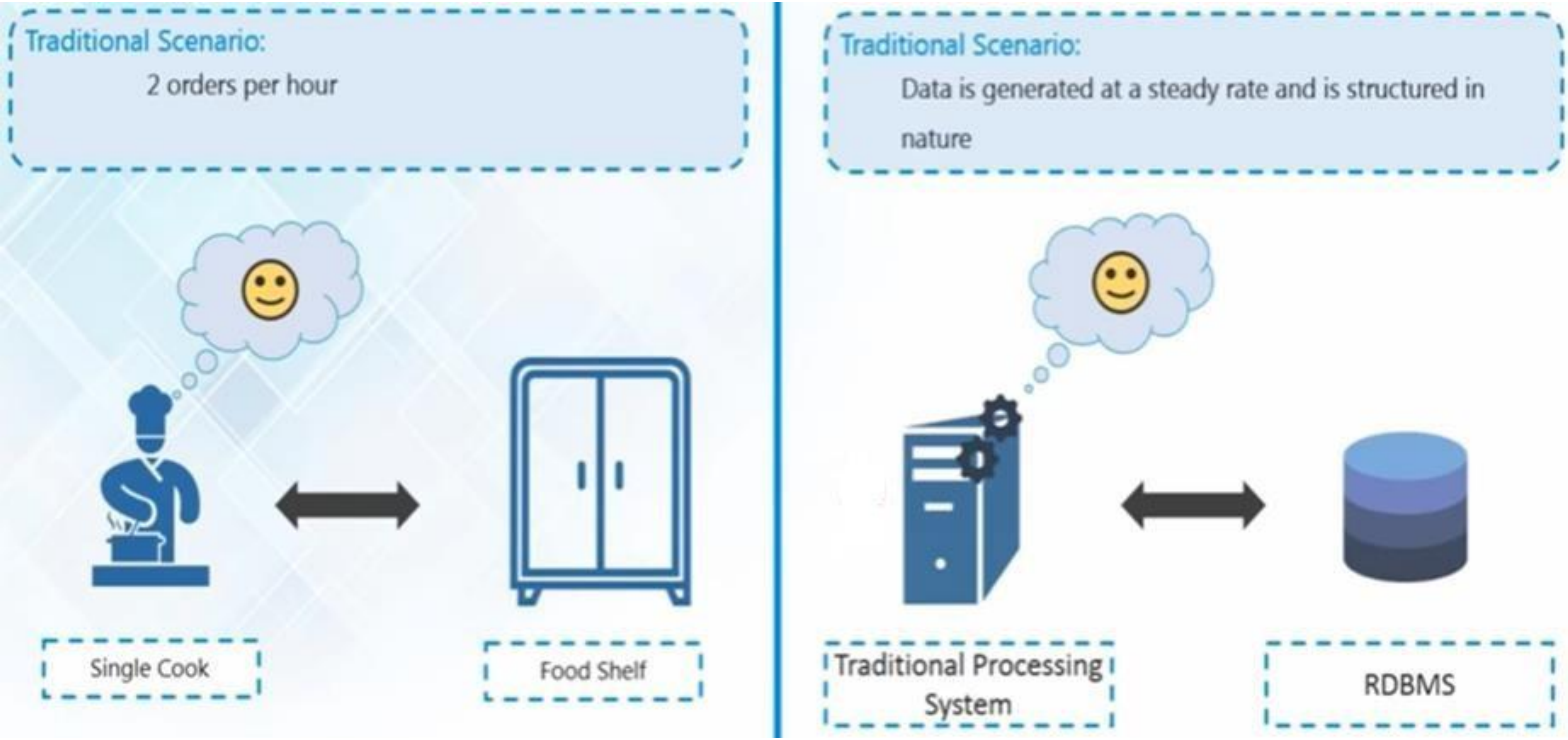
# Story of Big Data and Traditional System

Scenario:
Bob has opened a small restaurant in his city

# Traditional Svstem



# Traditional Scenario



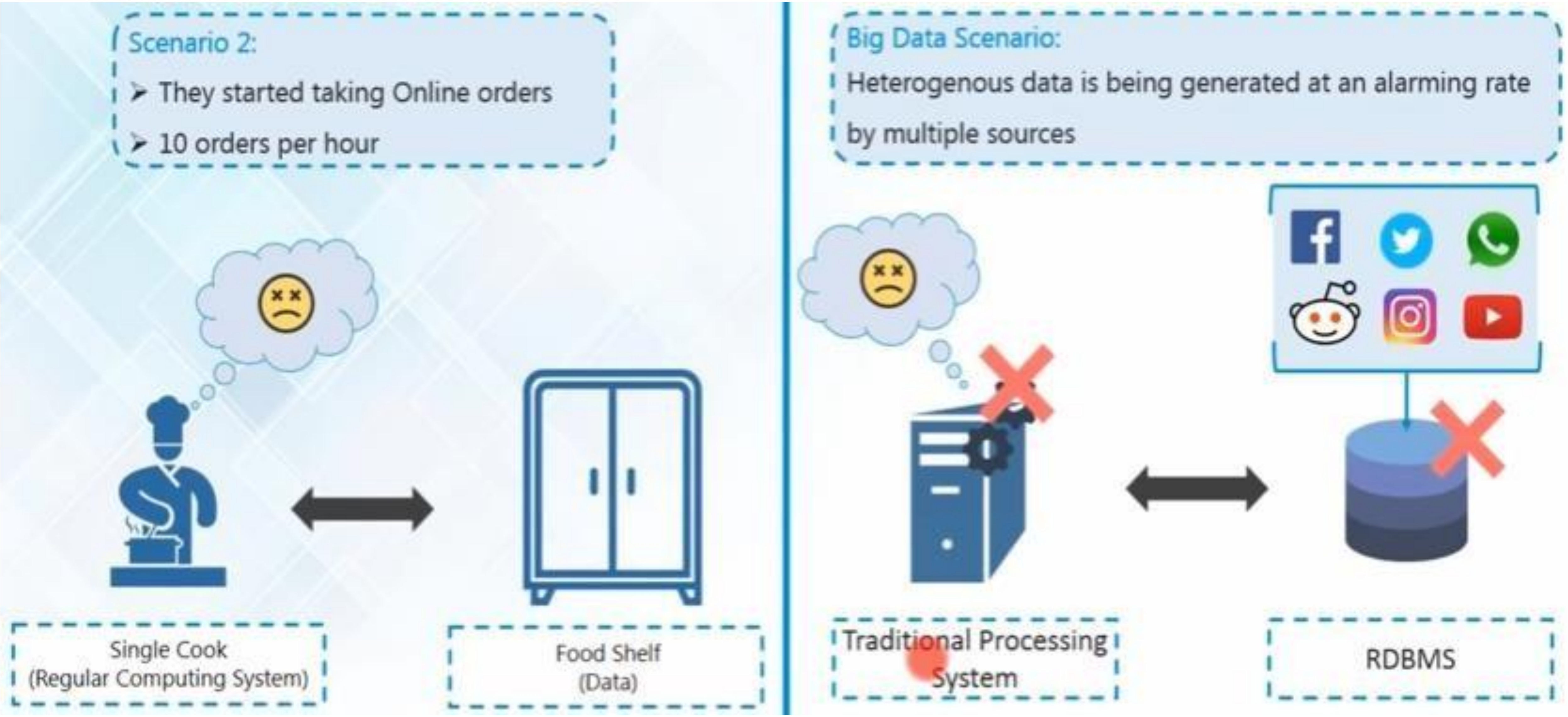
Traditional Scenario:

2 orders per hour

Traditional Scenario:

Data is generated at a steady rate and is structured in nature

# Failure of Traditional System



Scenario 2:

- They started taking Online orders
- 10 orders per hour

Single Cook (Regular Computing System)

Food Shelf (Data)

Big Data Scenario:

Heterogenous data is being generated at an alarming rate by multiple sources

Traditional Processing System

RDBMS

# Issue1 : Too many orders per  hour

Solution : Hire Multiple Cooks

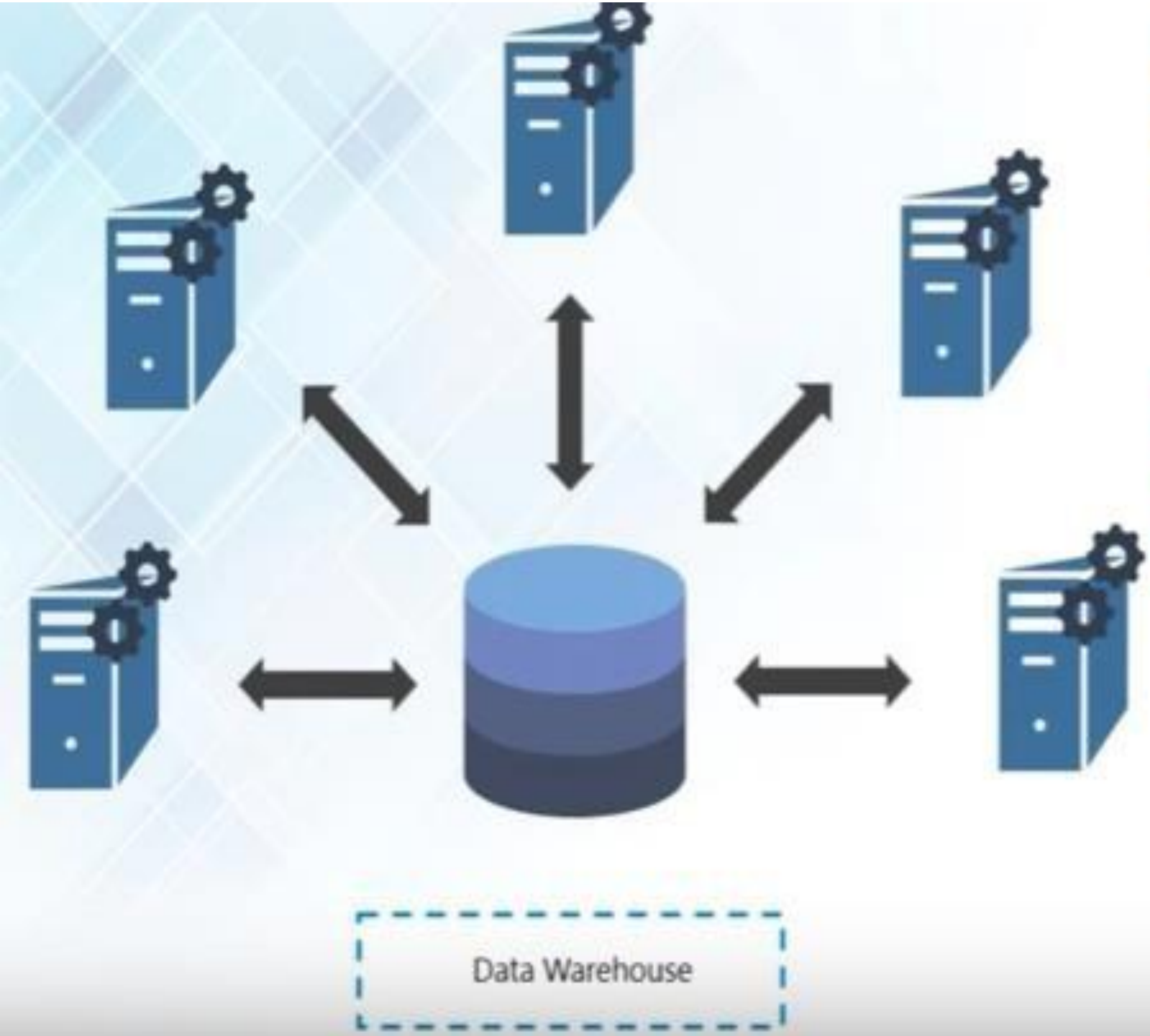
Food Shelf
(Data)

Scenario:

Multiple Cook cooking food

Issue:

Food Shelf becomes the BOTTLENECK



Scenario:

Multiple Processing Unit for data processing

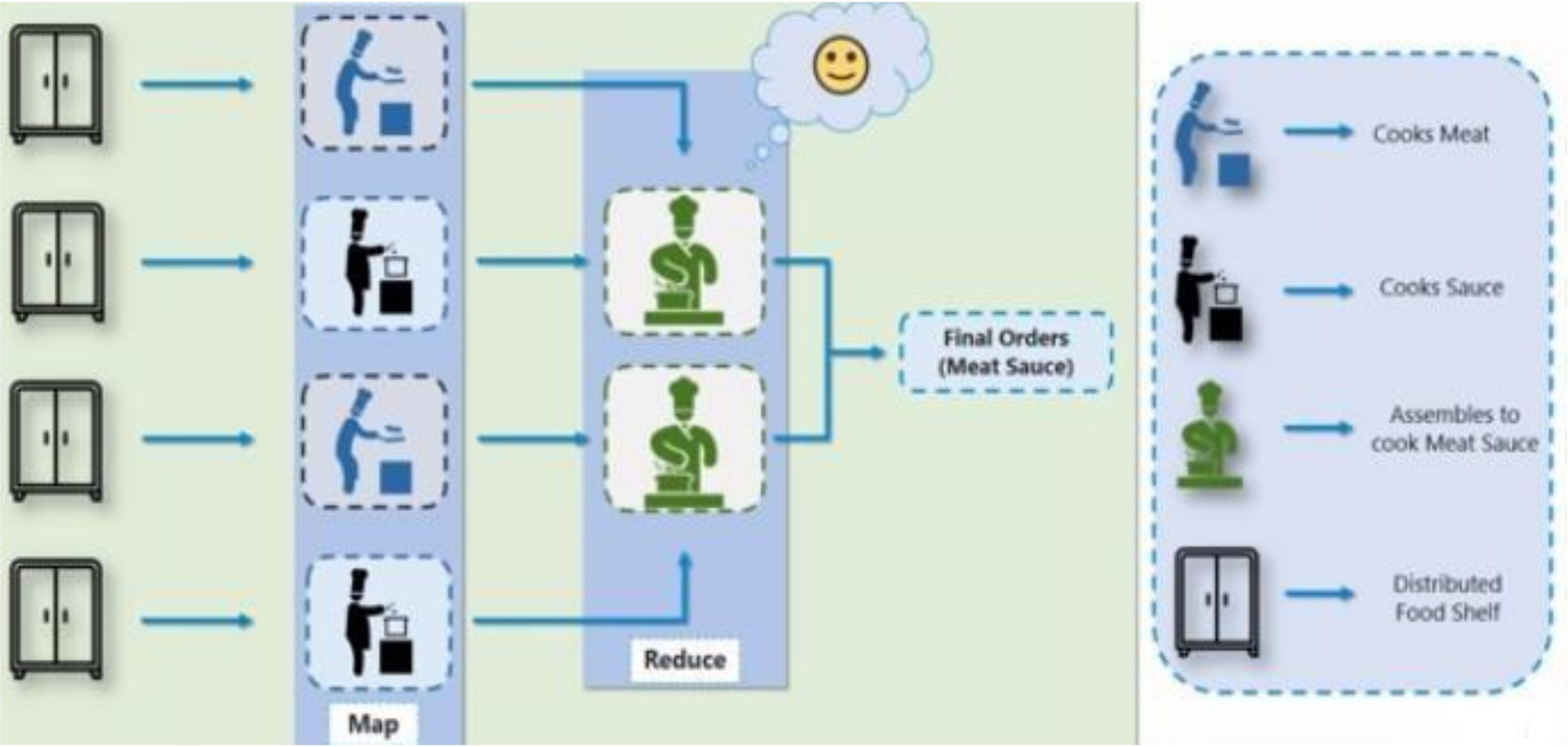
Issue:

Bringing data to processing generated lots of Network overhead

# Issue 2 : Food shelf becomes the bottleneck

## Solution : Distributed and Parallel approach

# Effective Solution



# Need a Framework

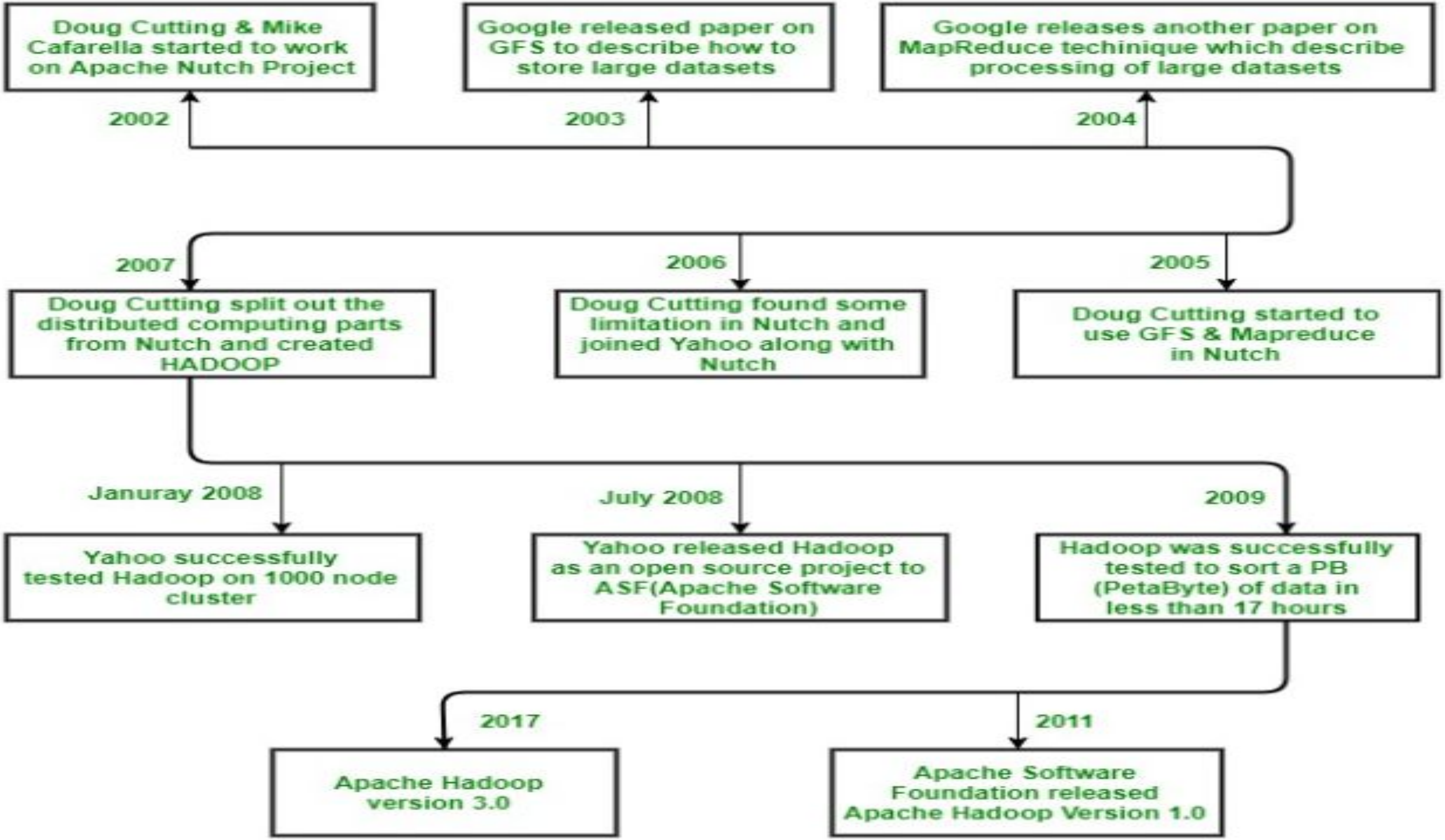
# Apache Hadoop:
# Framework to process Big data

# What is Hadoop?

- developed by Doug cutting

- Apache Hadoop is an open-source framework based on Google’s file system that can deal with big data in a distributed environment. This distributed environment is built up of a cluster of machines that work closely together to give an impression of a single working machine.

- **Why the hadoop name is given?**



2002

Doug Cutting & Mike Cafarella started to work on Apache Nutch Project

2003

Google released paper on GFS to describe how to store large datasets

2004

Google releases another paper on MapReduce techinique which describe processing of large datasets

2005

Doug Cutting started to use GFS & Mapreduce in Nutch

2006

Doug Cutting found some limitation in Nutch and joined Yahoo along with Nutch

2007

Doug Cutting split out the distributed computing parts from Nutch and created HADOOP

Januray 2008

Yahoo successfully tested Hadoop on 1000 node cluster

July 2008

Yahoo released Hadoop as an open source project to ASF(Apache Software Foundation)

2009

Hadoop was successfully tested to sort a PB (PetaByte) of data in less than 17 hours

2011

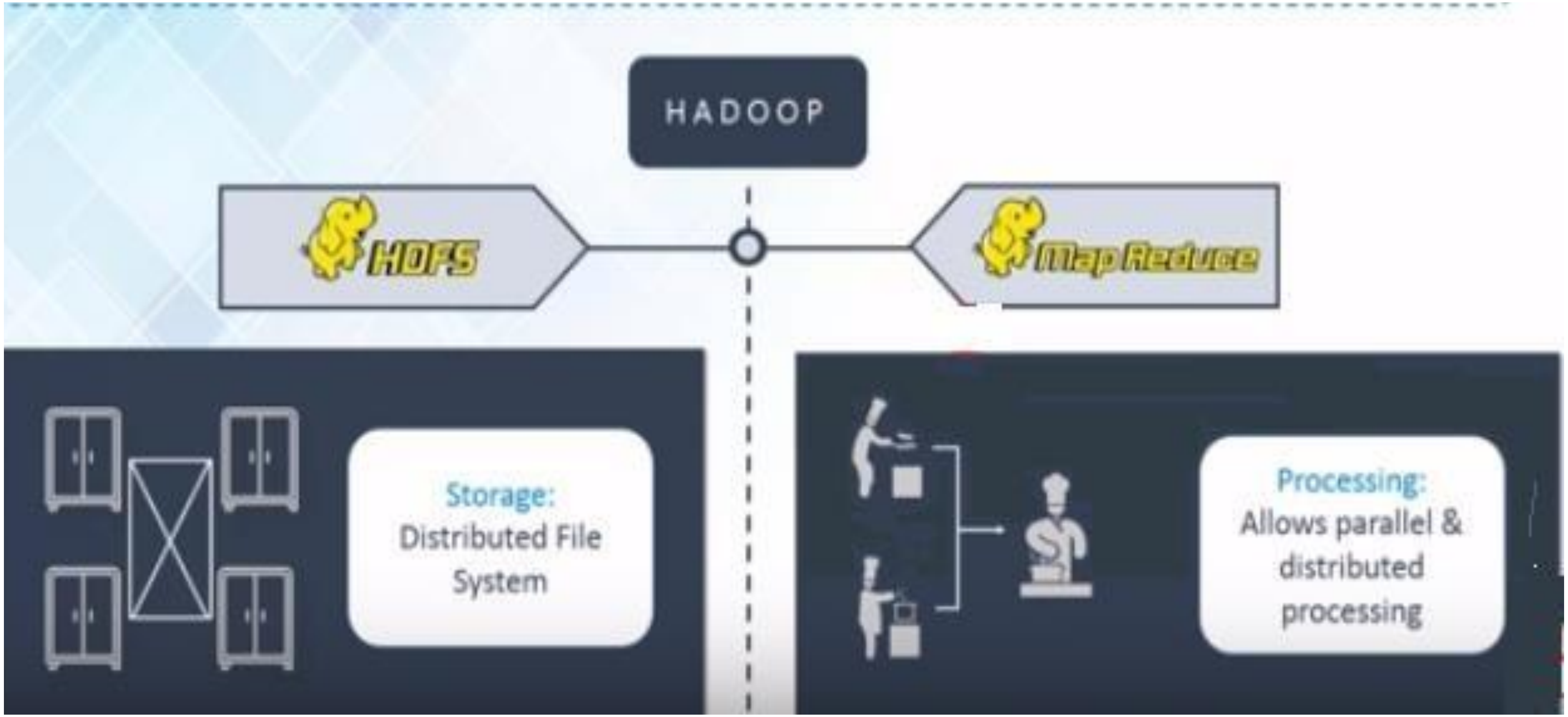
Apache Software Foundation released Apache Hadoop Version 1.0

2017

Apache Hadoop version 3.0

# Apache Hadoop:
# Framework to process Big data

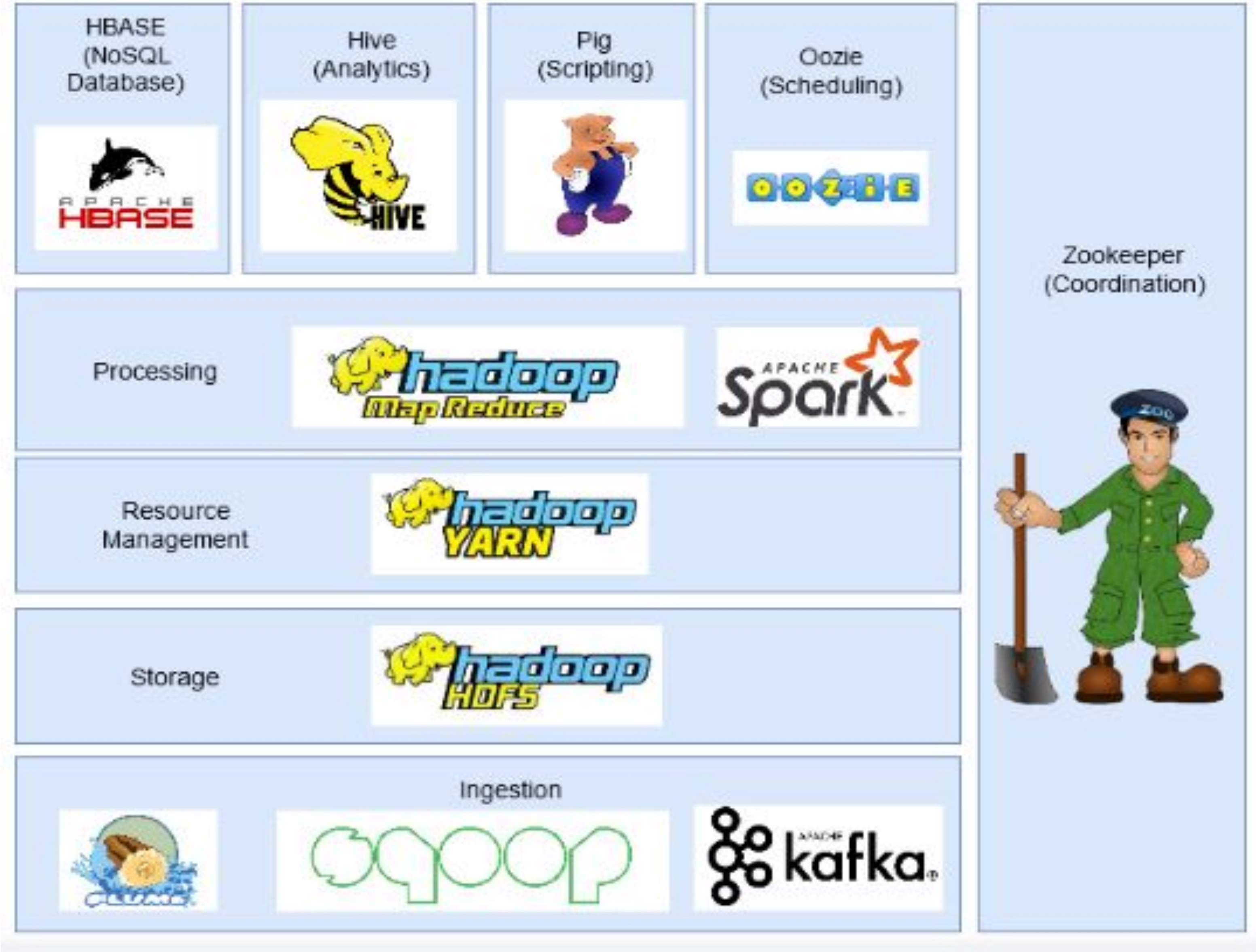
Hadoop is a Framework that allows us to store and process large data sets in parallel and distributed fashion.



# Advantages of Hadoop Framework

- Hadoop is **highly scalable** because it handles data in a distributed manner
- Compared to vertical scaling in RDBMS, Hadoop offers **horizontal scaling**
- It creates and saves replicas of data making it **fault-tolerant**
- It is **economical** as all the nodes in the cluster are commodity hardware which is nothing but inexpensive machines
- Hadoop utilizes the **data locality concept** to process the data on the nodes on which they are stored rather than moving the data over the network thereby reducing traffic
- It can **handle any type of data**: structured, semi-structured, and unstructured. This is extremely important in today's time because most of our data (emails, Instagram, Twitter, IoT devices, etc.) has no defined format

# Hadoop Ecosystem



# Components of the Hadoop Ecosystem

- **HDFS (Hadoop Distributed File System)**

  HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes

- **MapReduce**

  By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing logic and helps to write applications which transform big data sets into a manageable one.

- **YARN**

  Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.

- **HBase**

  It’s a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database.

- **Pig**

  Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.

- **Apache Spark**

  Apache Spark is an essential product from the Apache software foundation, and it is considered as a powerful data processing engine. Spark is empowering the big data applications around the world. It all started with the increasing needs of enterprises and where MapReduce is unable to handle them.

- **Hive**

  HIVE performs reading and writing of large data sets. However, its query language is called as HQL

- **Sqoop**

  a. Sqoop is a front-end interface that enables in moving bulk data from Hadoop to relational databases and into variously structured data marts.

- **Flume**

  Flume collects, aggregates, and moves large sets of data from its origin and sends it back to HDFS. It works as a fault-tolerant mechanism. It helps in transmitting data from a source into a Hadoop environment.

- **Kafta**

  Apache Kafka is a distributed streaming system that is emerging as the preferred solution for integrating real-time data from multiple stream-producing sources and making that data available to multiple stream-consuming systems concurrently

- **Oozie**

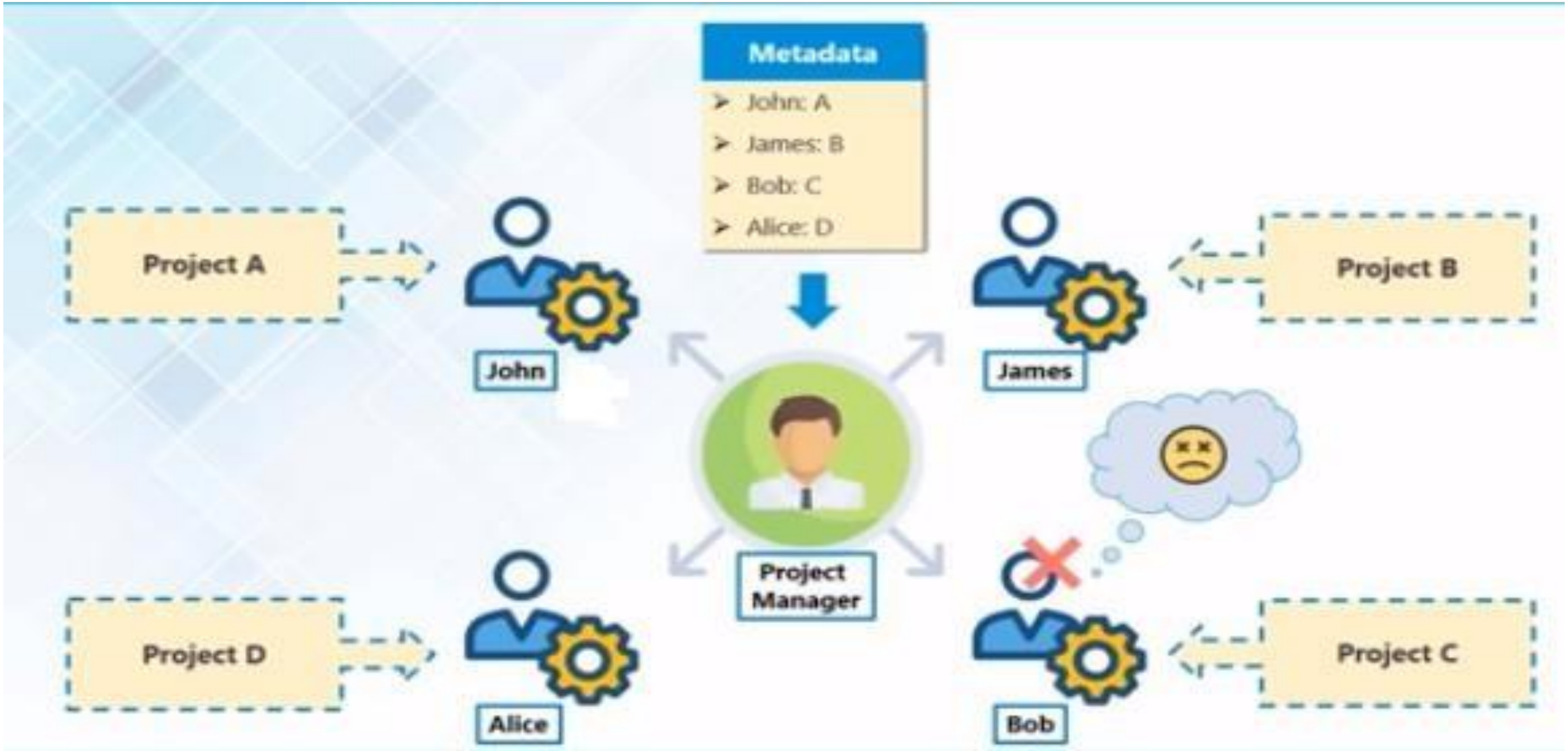
  Apache Ooze is a tool in which all sort of programs can be pipelined in a required manner to work in Hadoop's distributed environment. Oozie works as a  scheduler system to run and manage Hadoop jobs.

- **Zookeeper**

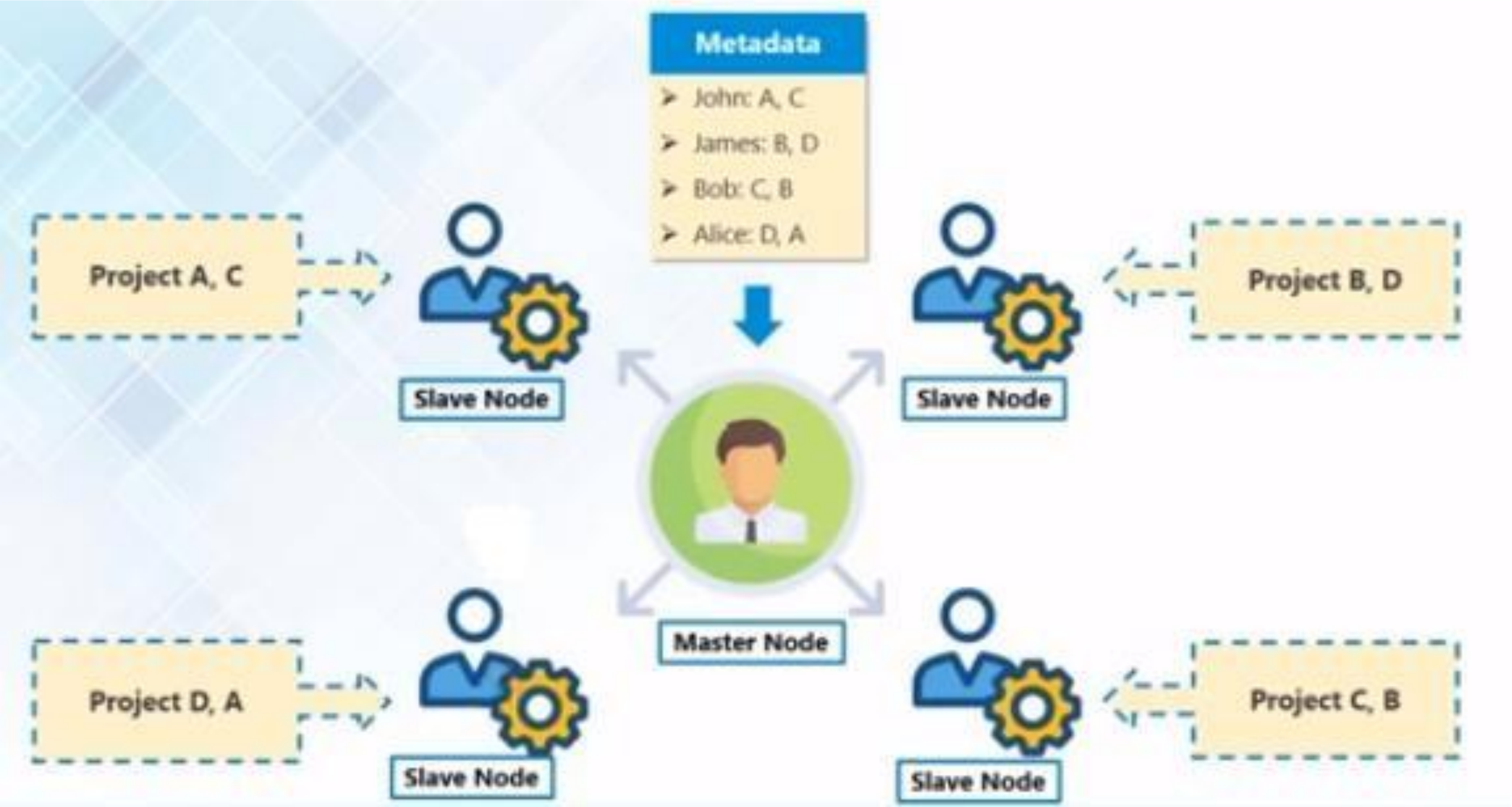
- Apache Zookeeper is an open-source project designed to coordinate multiple services in the Hadoop ecosystem. Organizing and maintaining a service in a distributed environment is a complicated task. Zookeeper solves this problem with its simple APIs and Architecture. Zookeeper allows developers to focus on core applications instead of concentrating on a distributed environment of the application.

# Hadoop: Master/Slave Architecture

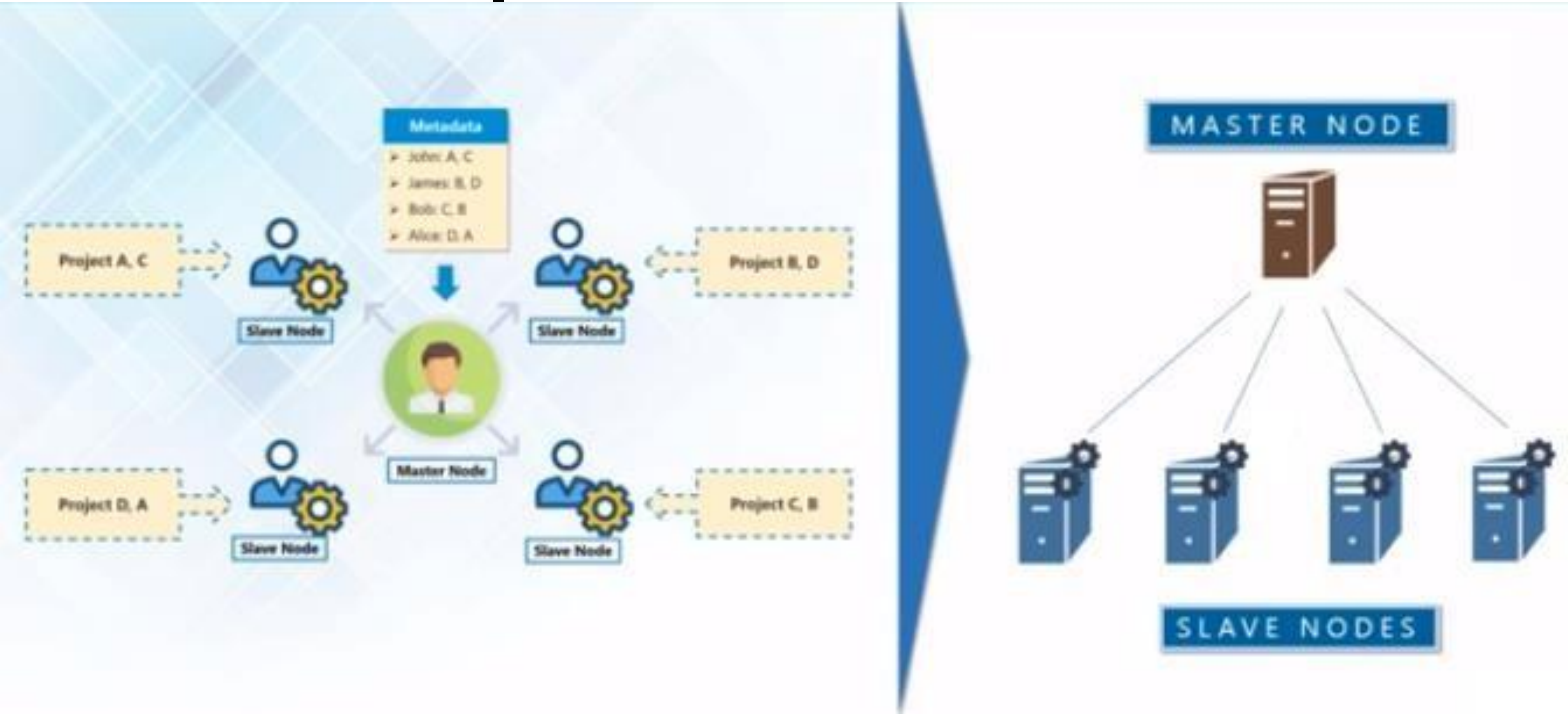
# Hadoop: Master/Slave Architecture



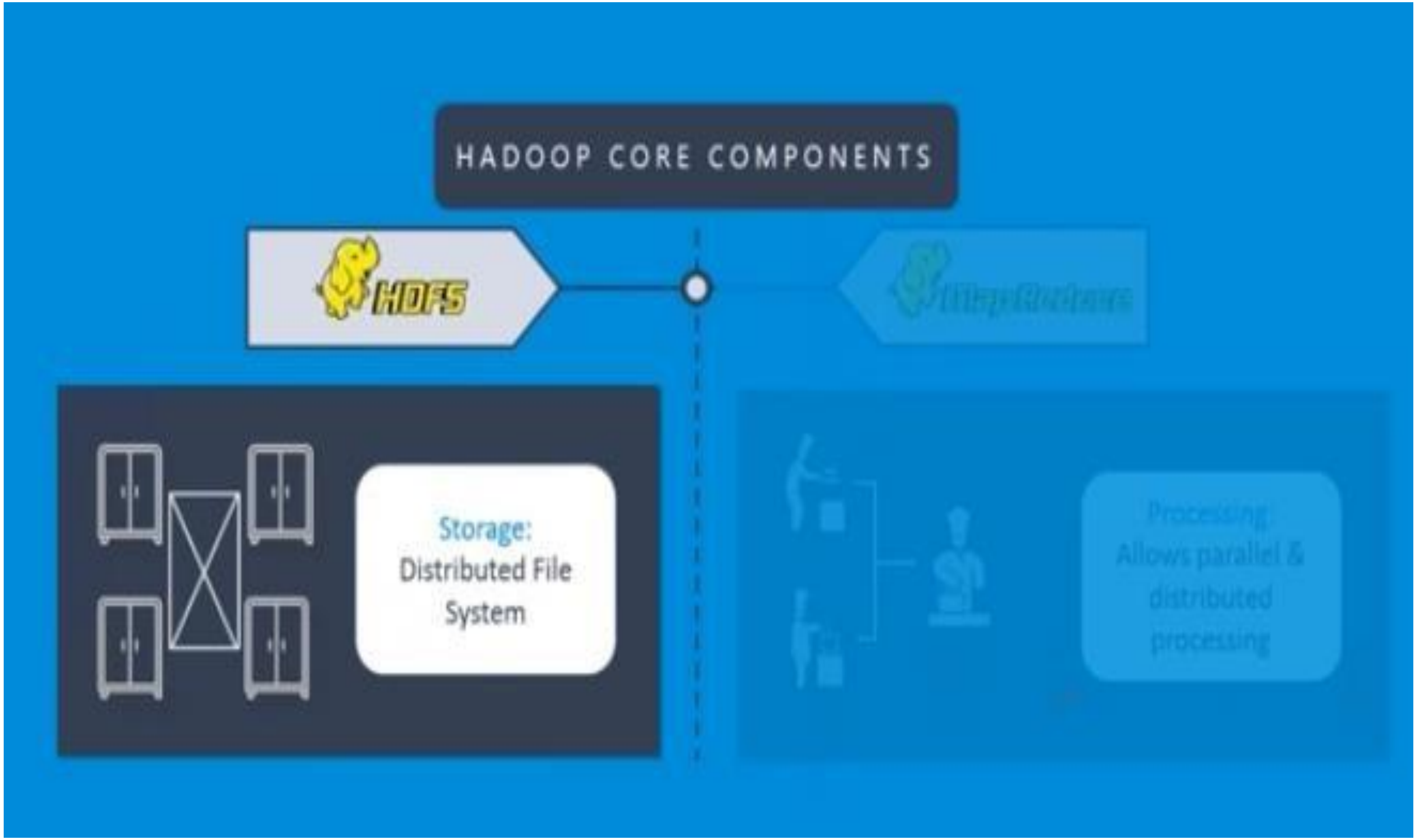
# Hadoop: Master/Slave Architecture



# Hadoop: Master/Slave Architecture



# HDFS



HADOOP CORE COMPONENTS

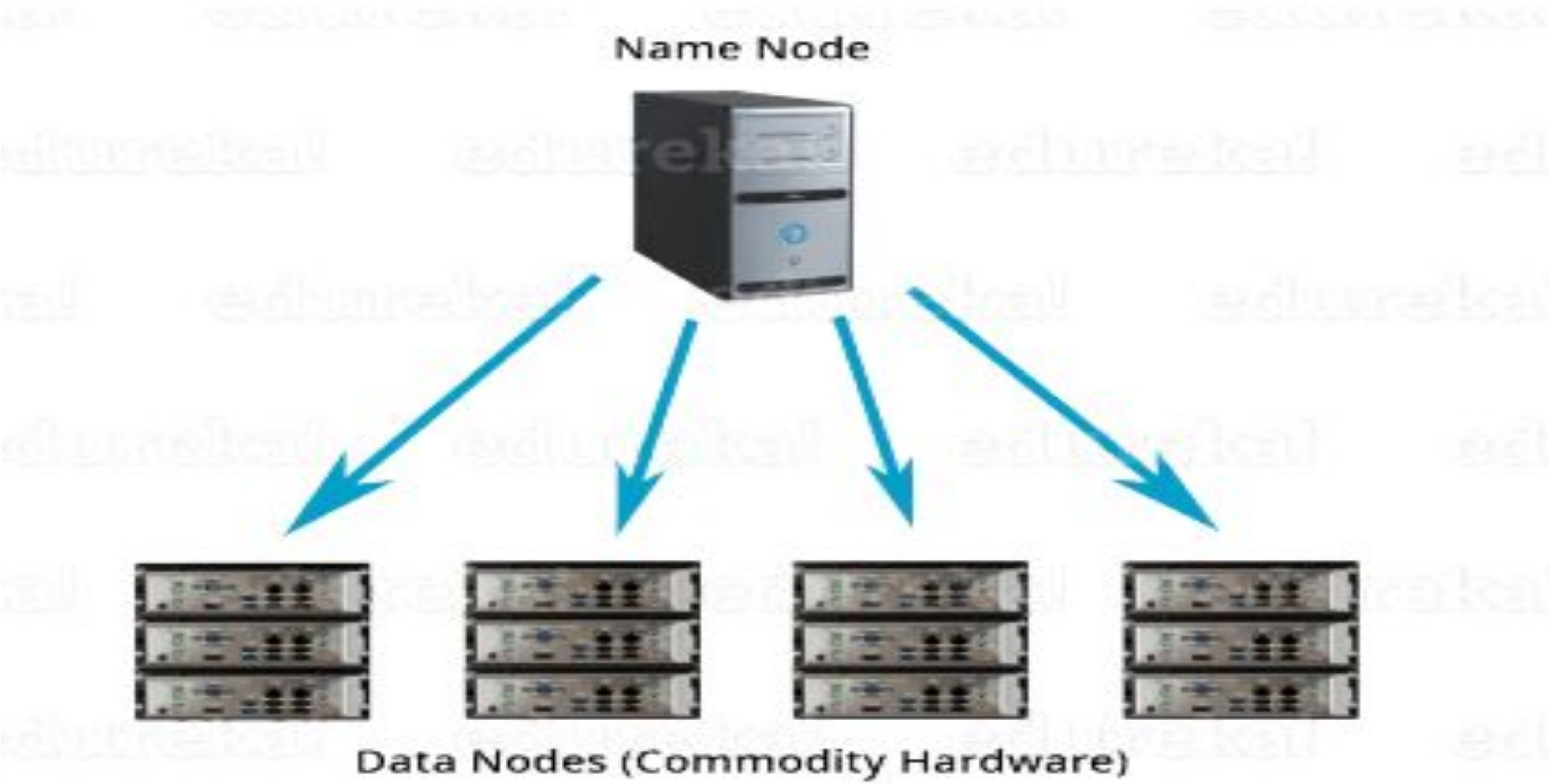
HDFS

MapReduce

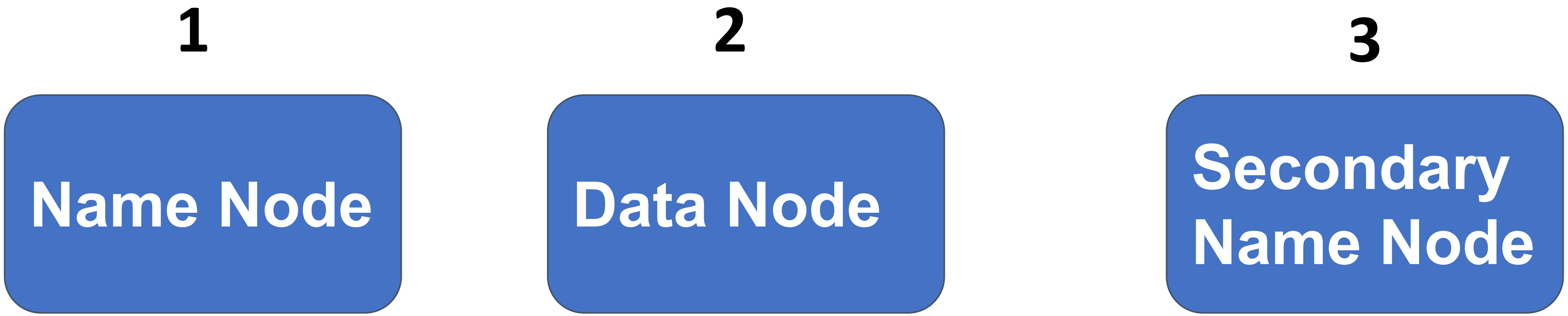
Storage:
Distributed File System

Processing:
Allows parallel & distributed processing

# “Moving Computation is Cheaper than Moving Data”



# HDFS Core Components



1

Name Node

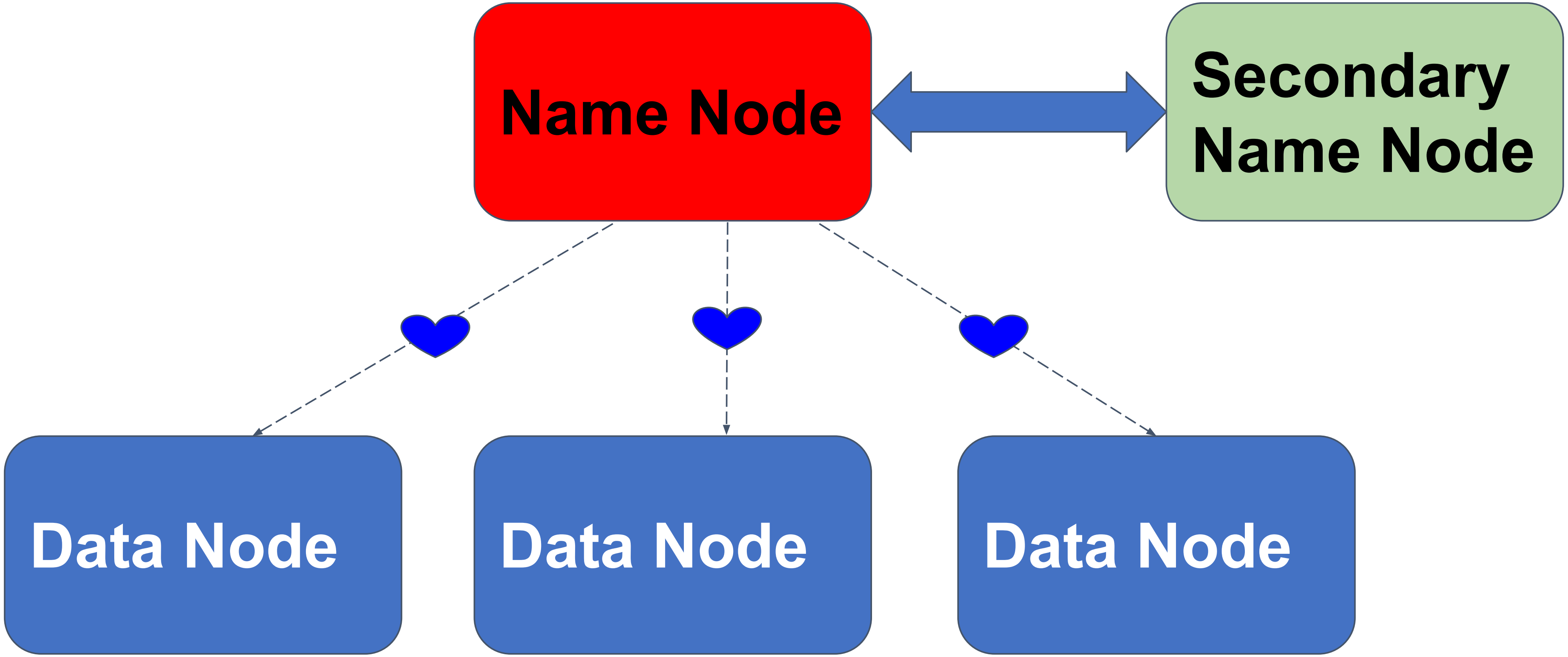
2

Data Node

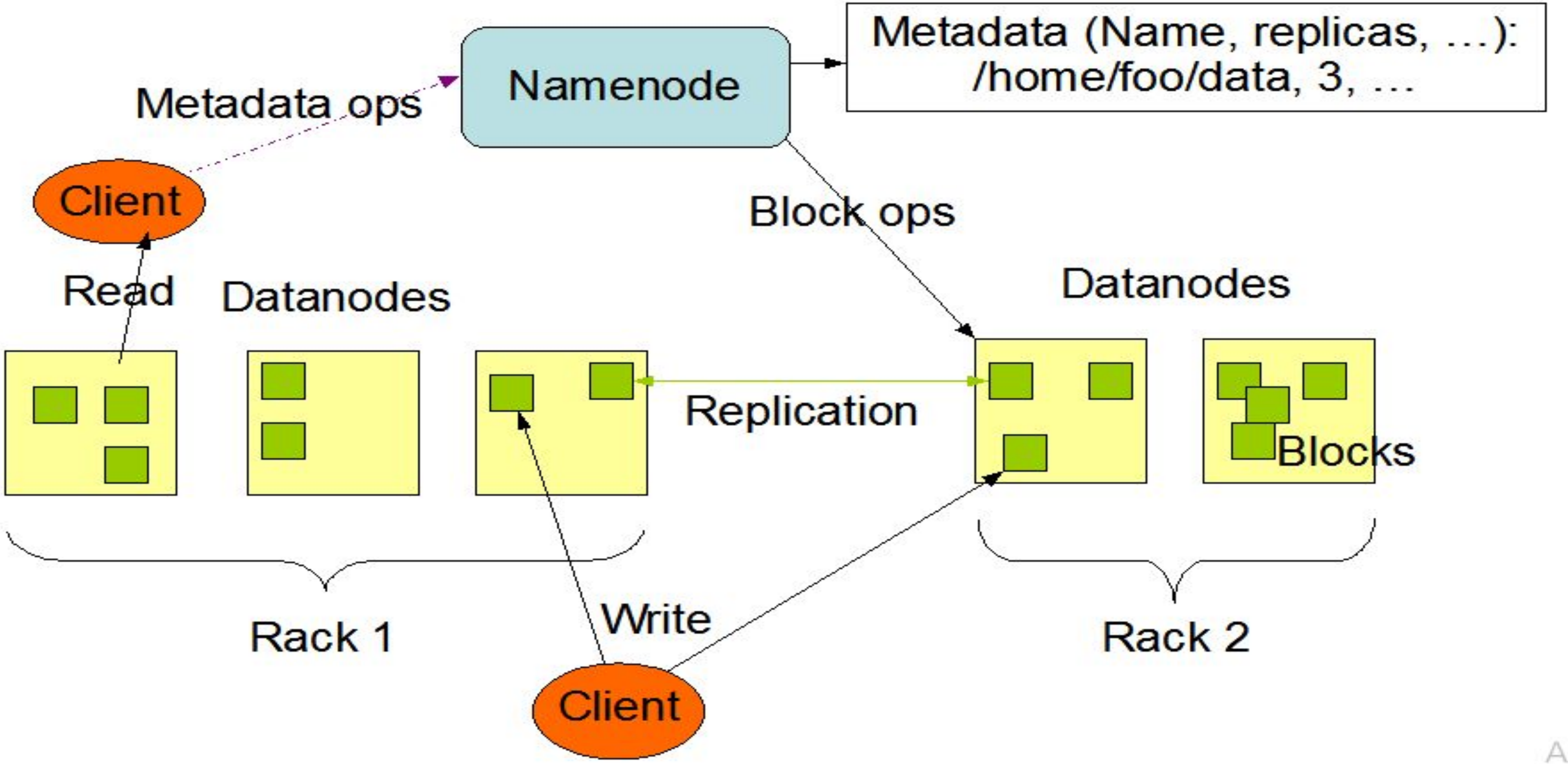
3

Secondary Name Node

# NameNode and DataNode



HDFS Architecture



https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

# Name Node

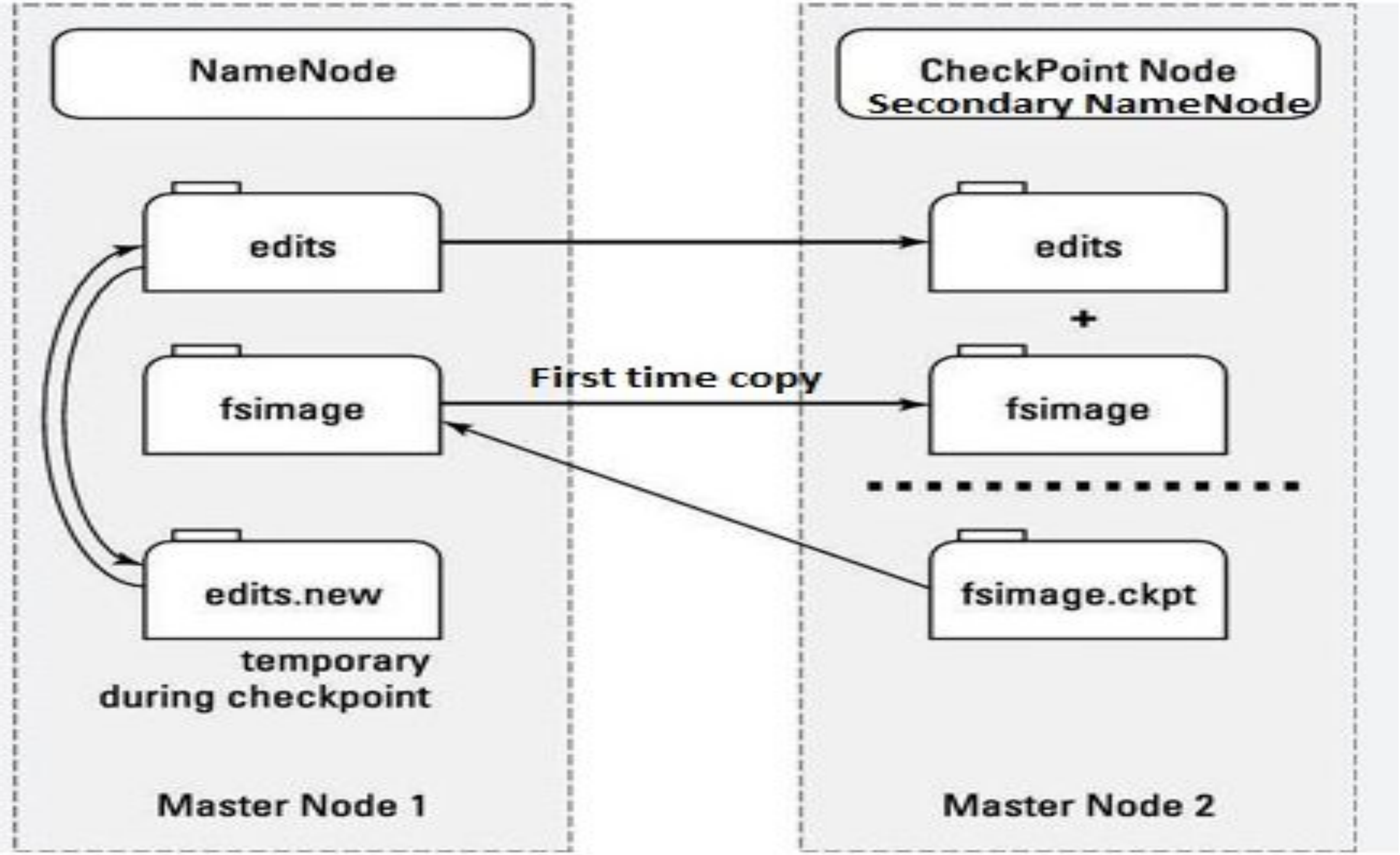
- Maintains and manages Datanodes
- Records metadata i.e . information about the data blocks (e.g.location of the block stored, the size of file, permissions, hierarchy etc.
- Received heartbeat
- store block report from all data nodes

# DataNode

- Slave daemons
- stores actual data
- serves read and write requests from the clients

# Secondary Namenode & Checkpointing

editLog(in RAM) stores recent changes in RAM, fsImage(in Disk) stores all changes in Disk



# Checkpointing

- Checkpointing is the process of combining FsImage with Editlogs.
- Secondary Namenode take over the responsibility of checkpointing ,therefore making namenode more available.
- prevents EditLogs from getting too large.
- checkpointing happens periodically(default : 1 hour)

# How the data is actually stored in Datanodes?

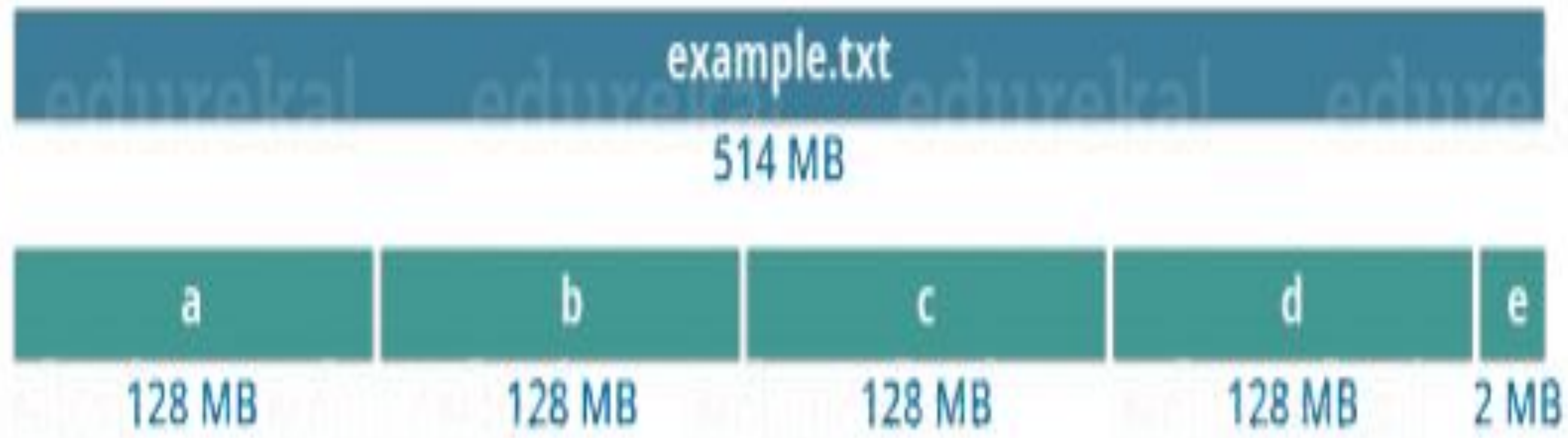
# HDFS Data blocks

# HDFS

- A Hadoop cluster is a collection of computers, known as nodes, that are networked together to perform parallel computations on big data sets.
- Divide the large file into pieces called blocks.
- It uses rackId to identify the datanode from rack .
- A rack is the collection of datanodes within the cluster.

# HDFS Data Blocks

- Blocks are the nothing but the smallest continuous location on your hard drive where data is stored.
- In general, in any of the File System, you store the data as a collection of blocks. Similarly, HDFS stores each file as blocks which are scattered throughout the Apache Hadoop cluster.
- The default size of each block is 1**28 MB** in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x) which you can configure as per your requirement.

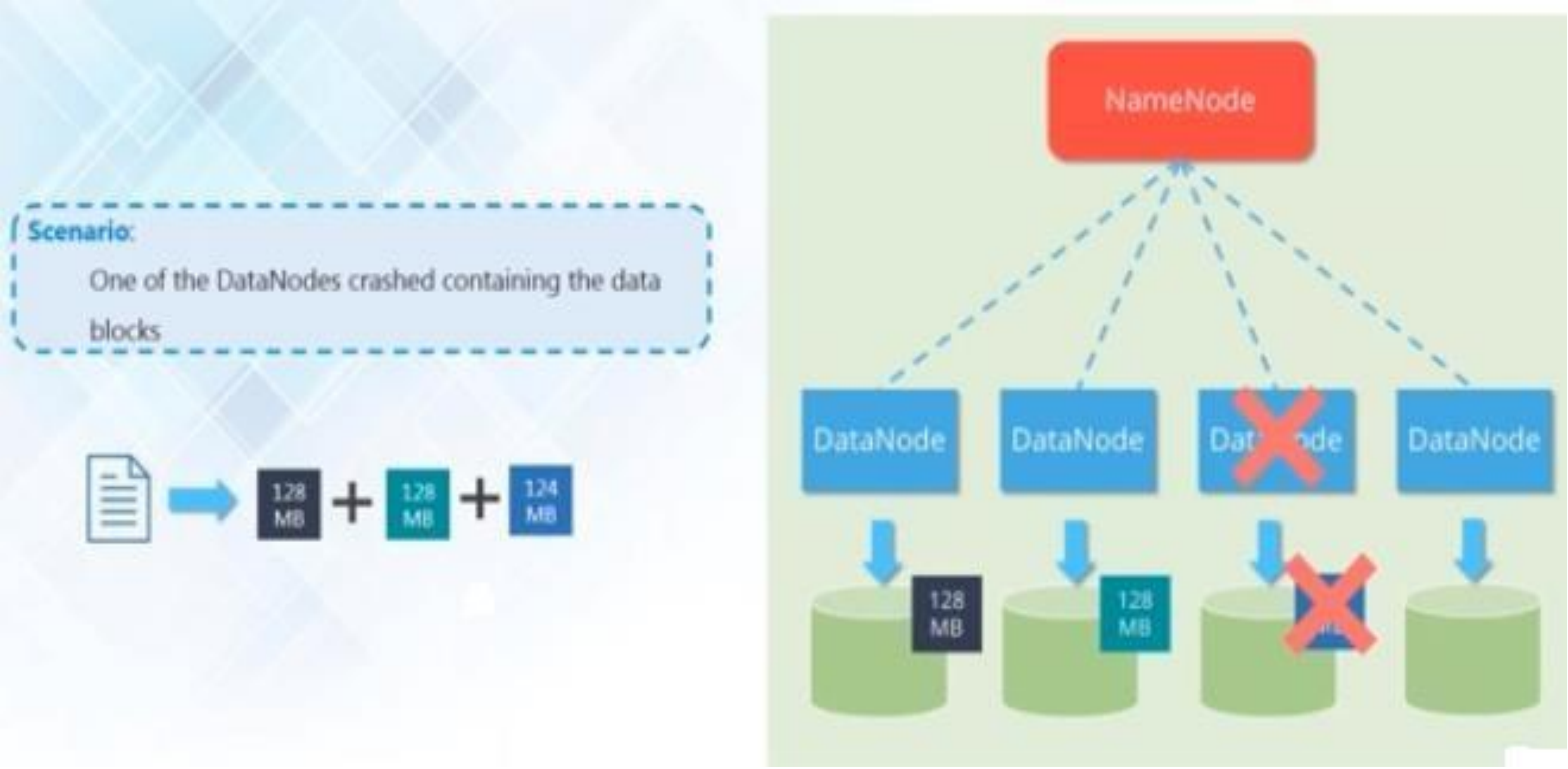


# Advantages of HDFS

- If data is more than the total size of a Hadoop cluster, more computers can be added to cluster.
- If a huge data file is stored on a single machine and it takes 4 seconds to process that file, in Hadoop cluster that data file can be divided into cluster and processing time can be reduced.

# Fault Tolerance:
# How Hadoop  cope up with Datanode failure?

# Fault Tolerance

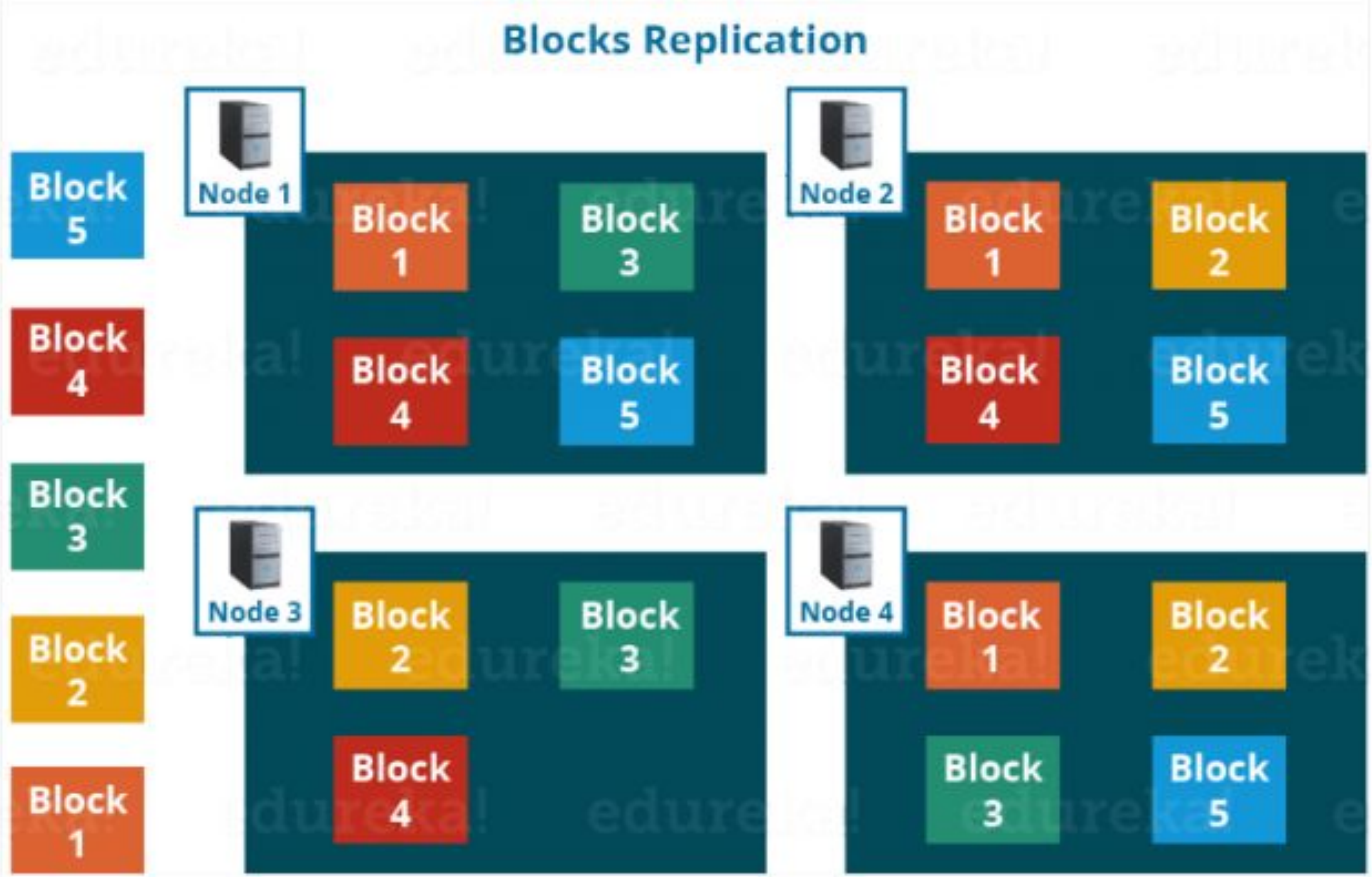


**Scenario:**

One of the DataNodes crashed containing the data blocks

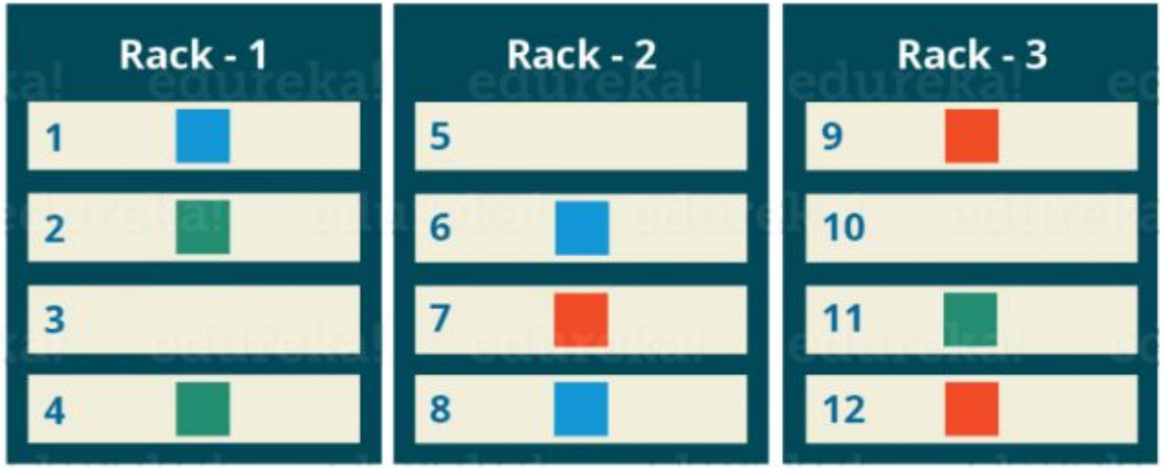
# Solution : Replication factor

# Fault Tolerance: Replication Factor

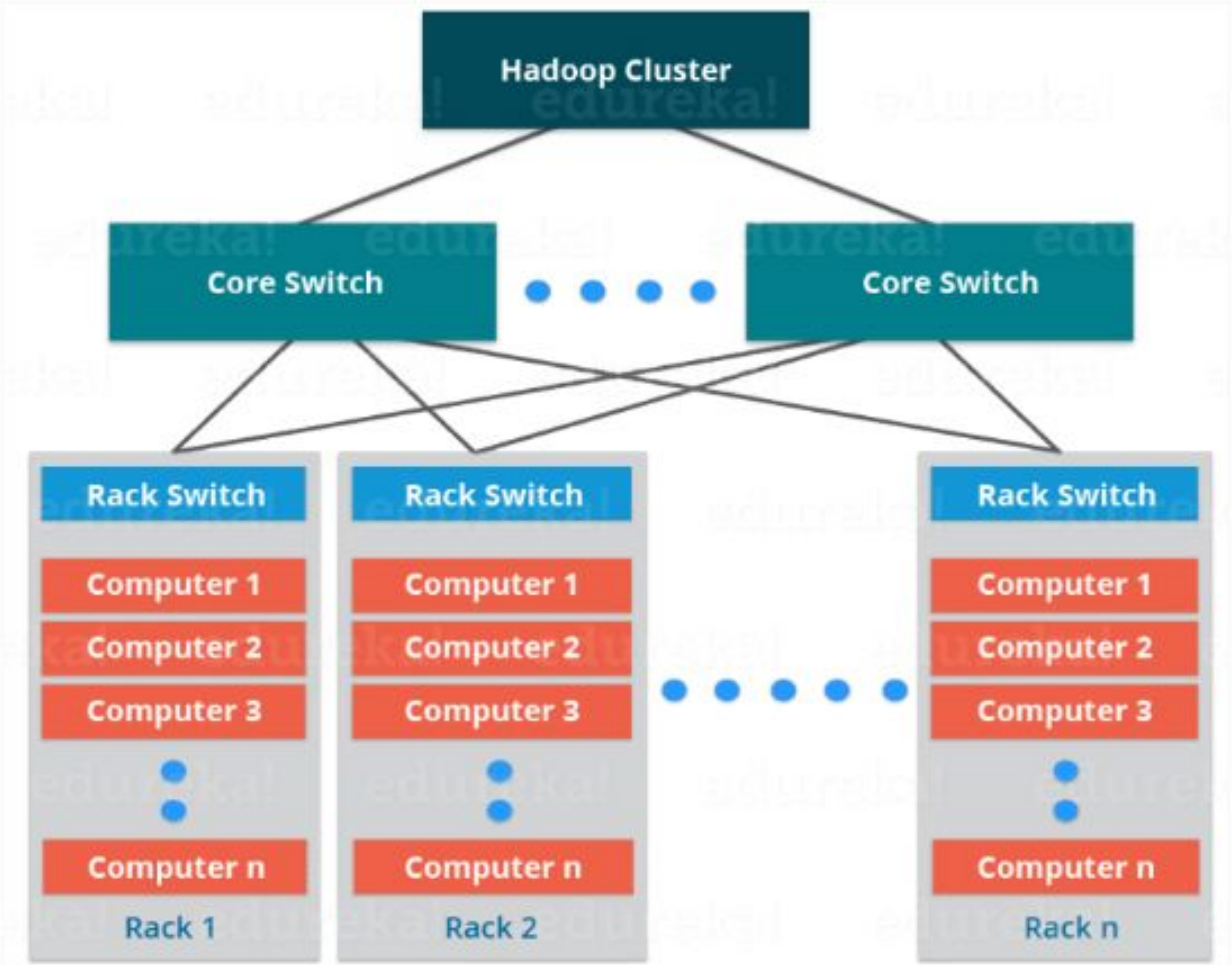


# Rack Awareness

- Hadoop components are rack-aware. For example, HDFS block placement will use rack awareness for fault tolerance by placing one block replica on a different rack. This provides data availability in the event of a network switch failure or partition within the cluster.
- Hadoop master daemons obtain the rack id of the cluster slaves by invoking either an external script or java class as specified by configuration files. Using either the java class or external script for topology, output must adhere to the java **org.apache.hadoop.net.DNSToSwitchMapping** interface.
- The interface expects a one-to-one correspondence to be maintained and the topology information in the format of '/myrack/myhost', where '/' is the topology delimiter, 'myrack' is the rack identifier, and 'myhost' is the individual host. Assuming a single /24 subnet per rack, one could use the format of '/192.168.100.0/192.168.100.5' as a unique rack-host topology mapping.



[https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-common/RackAwareness.html](https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-common/RackAwareness.html)

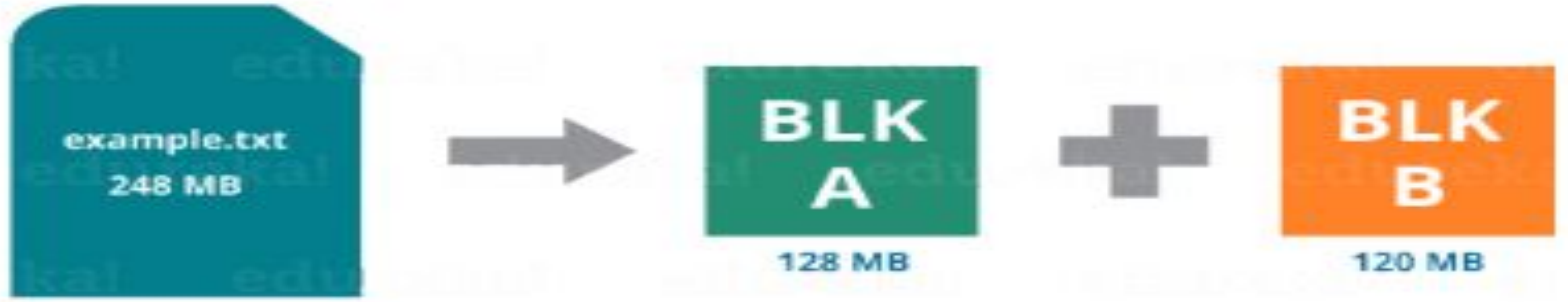

Hadoop Cluster
Core Switch
Core Switch
Rack Switch
Computer 1
Computer 2
Computer 3
Computer n
Rack 1
Rack Switch
Computer 1
Computer 2
Computer 3
Computer n
Rack 2
Rack Switch
Computer 1
Computer 2
Computer 3
Computer n
Rack n

# HDFS Write Mechanism

Suppose a situation where an HDFS client, wants to write a file named “example.txt” of size 248 MB.

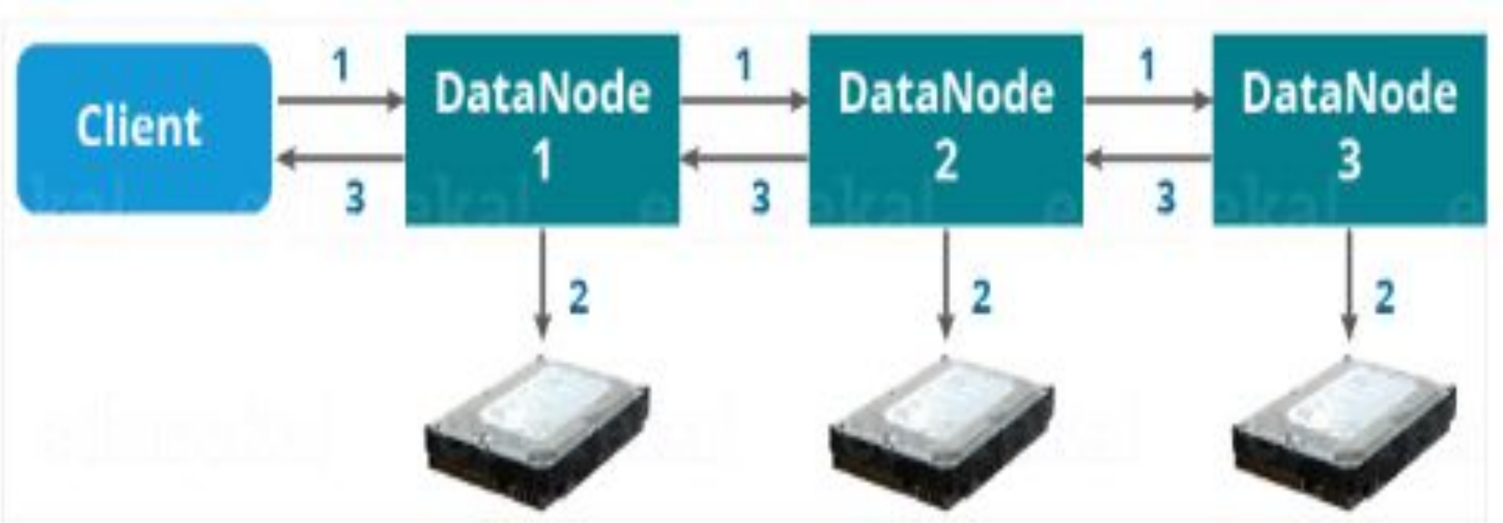
Assume that the system block size is configured for 128 MB (default). So, the client will be dividing the file “example.txt” into 2 blocks – one of 128 MB (Block A) and the other of 120 MB (block B).

- Suppose, the NameNode provided following lists of IP addresses to the client:
  - For Block A, list A = {IP of DataNode 1, IP of DataNode 4, IP of DataNode 6}
  - For Block B, set B = {IP of DataNode 3, IP of DataNode 7, IP of DataNode 9}

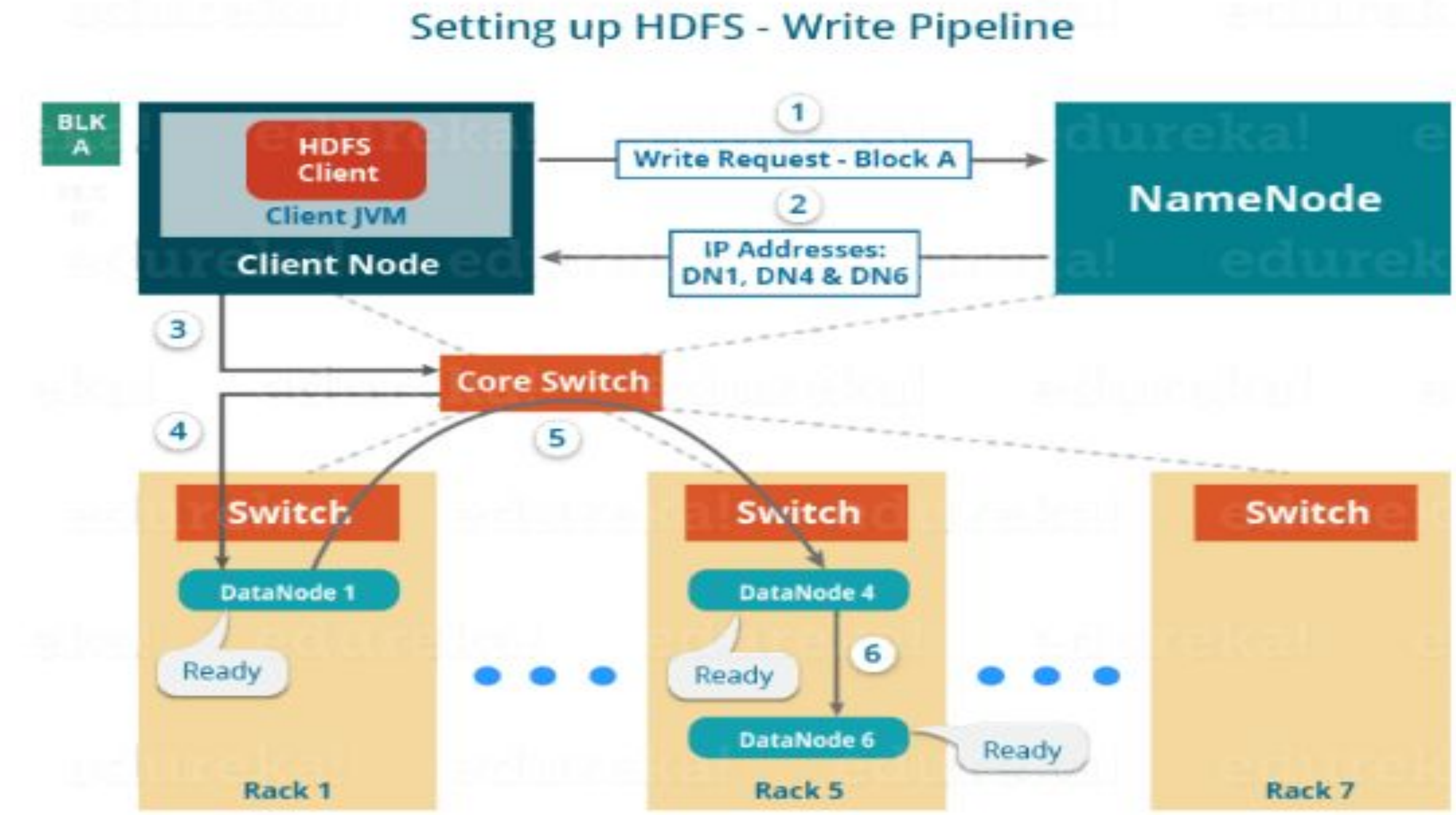


# HDFS Write Mechanism

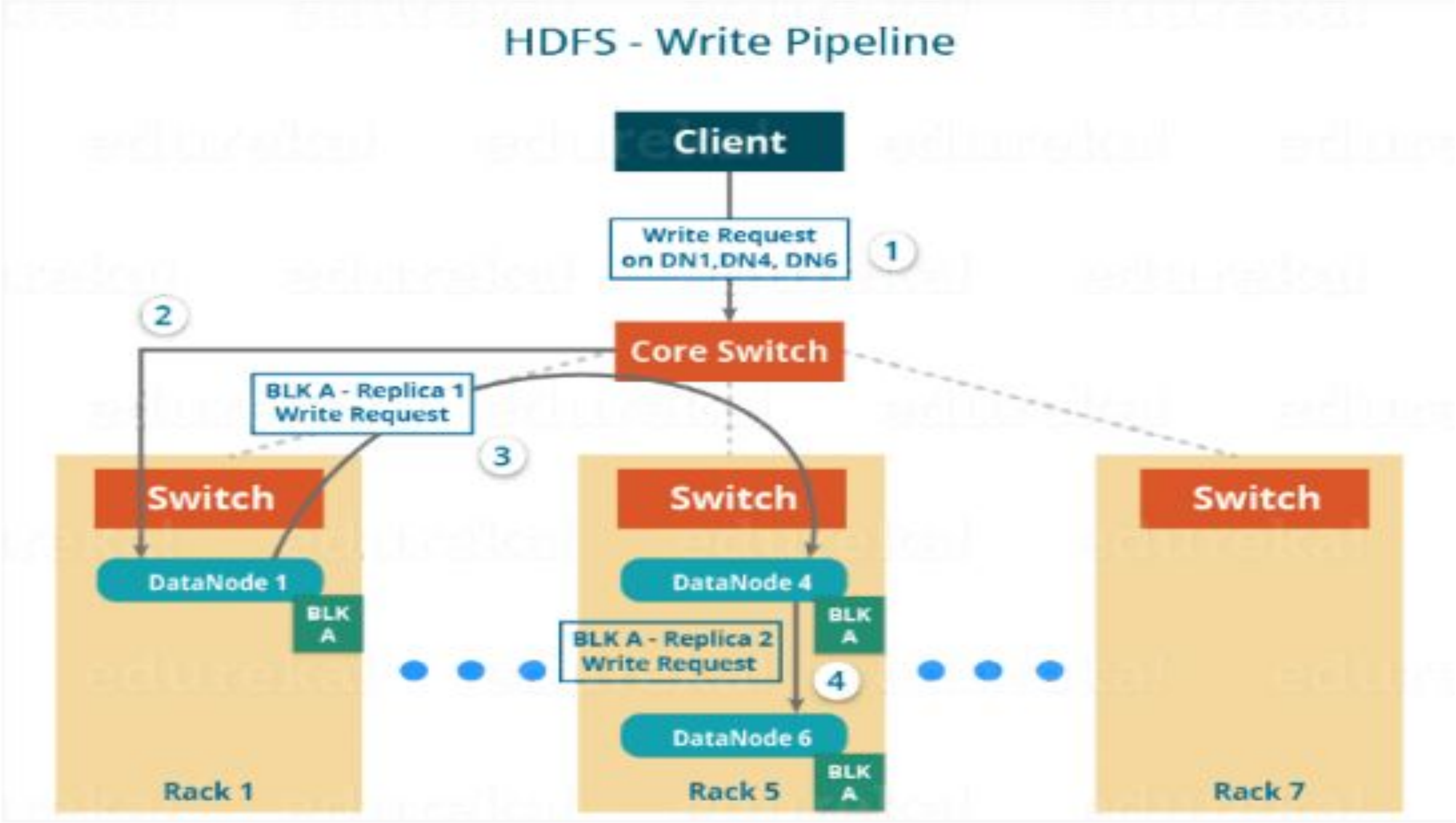
1. Set up of Pipeline
2. Data streaming and replication
3. Shutdown of Pipeline (Acknowledgement stage)



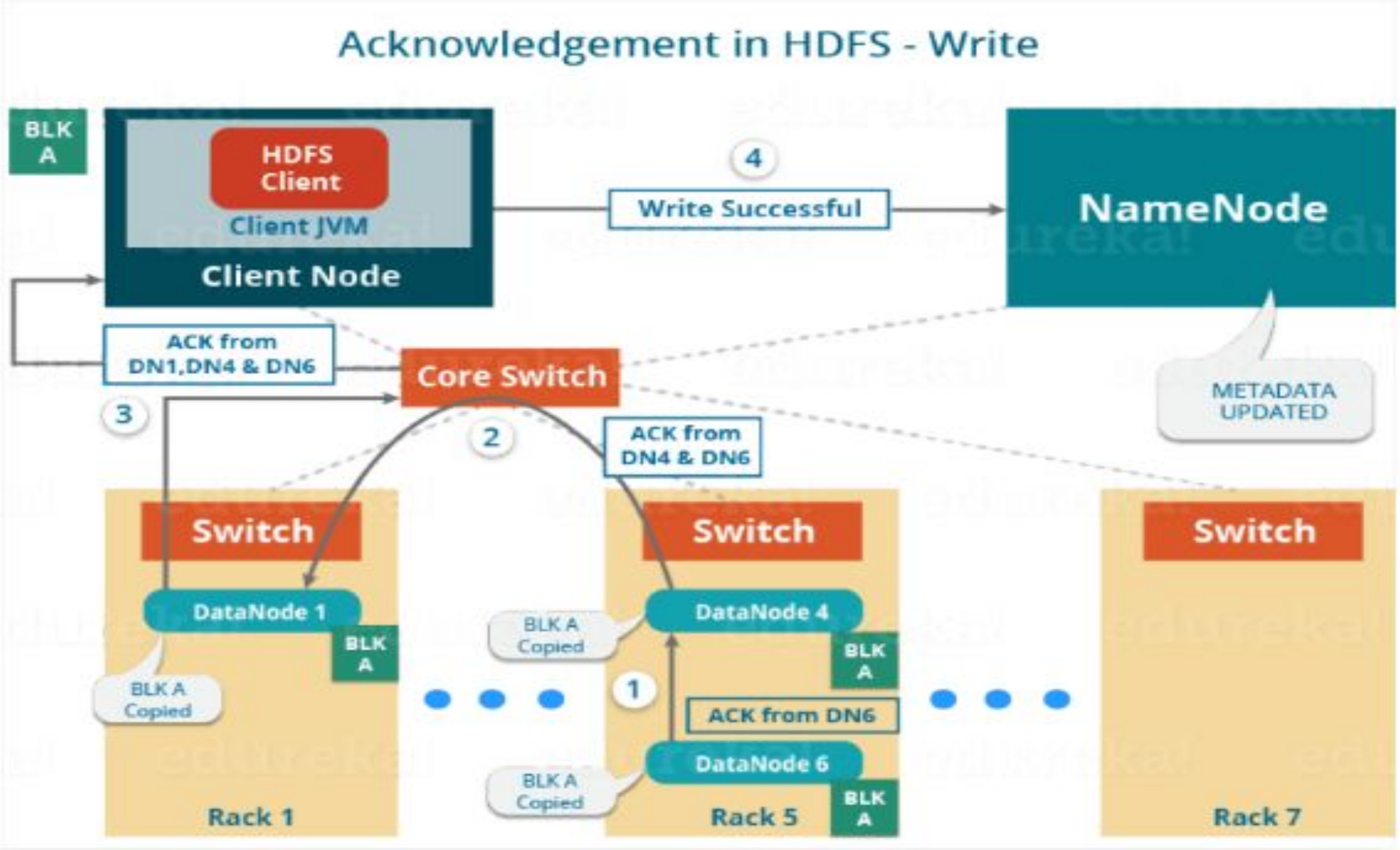
# HDFS Write Mechanism – Pipeline Setup



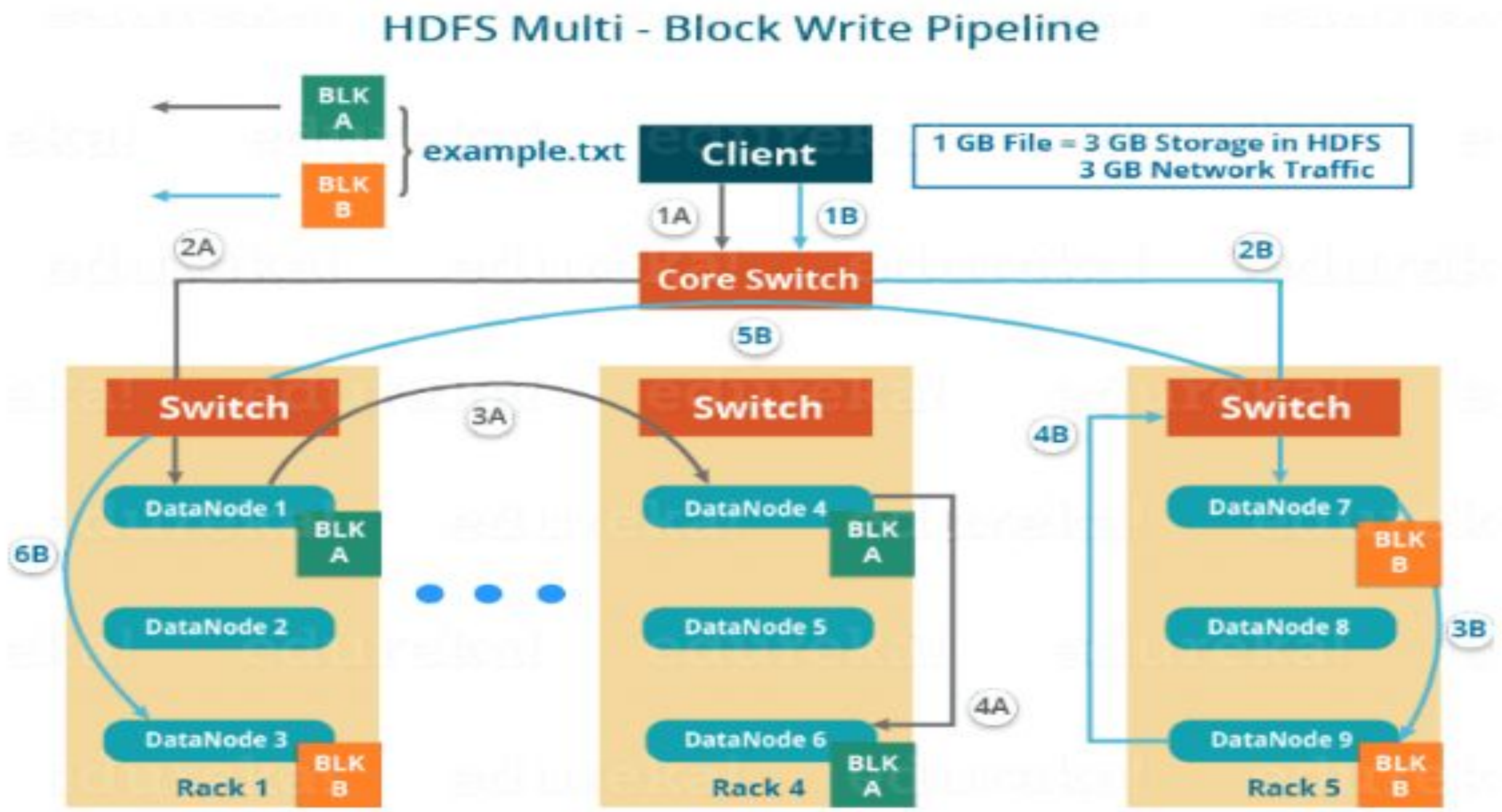
# HDFS Write Mechanism –Data Streaming



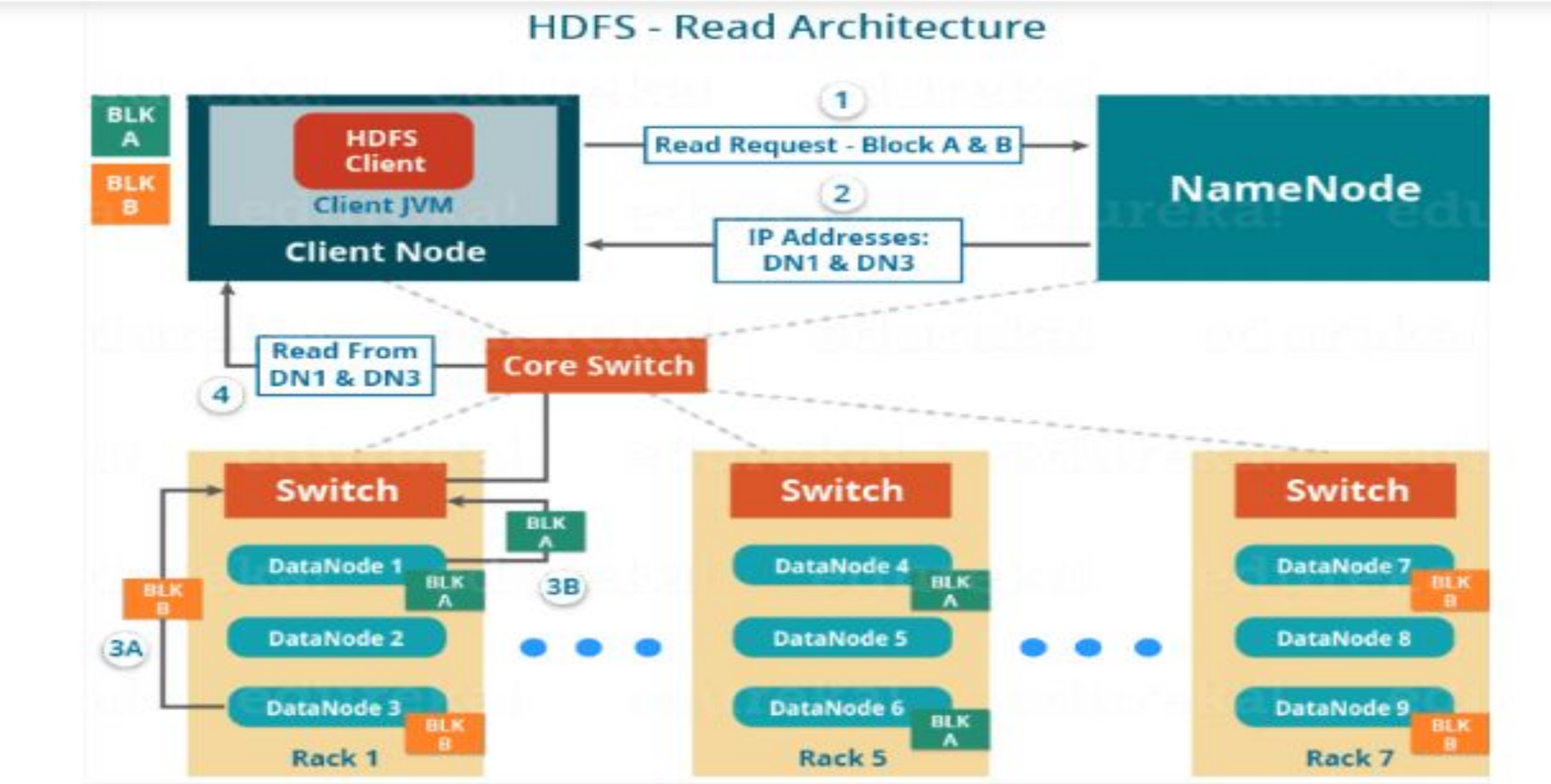
# HDFS Write Mechanism – Acknowledgment



# HDFS Multi-block Write Mechanism



# HDFS Read(Anatomy of File Read)



# Replica Selection

- To minimize global bandwidth consumption and read latency, HDFS tries to satisfy a read request from a replica that is **closest** to the reader.

- If there exists a replica on the same rack as the reader node, then that replica is preferred to satisfy the read request

- If HDFS cluster spans multiple data centers, then a replica that is resident in the local data center is preferred over any remote replica.

https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

## Assumptions and Goals of HDFS

- Hardware Failure
- Streaming Data Access
- Large Data Sets
- Simple Coherency Model

  (HDFS applications need a write-once-read-many access model for files)

- “Moving Computation is Cheaper than Moving Data”
- Portability Across Heterogeneous Hardware and Software Platforms

## The Communication Protocols

- All HDFS communication protocols are layered on top of the **TCP/IP protocol.**
- A client establishes a connection to a configurable TCP port on the NameNode machine.
- It talks the ClientProtocol with the NameNode.
- The DataNodes talk to the NameNode using the DataNode Protocol. A Remote Procedure Call (RPC) abstraction wraps both the Client Protocol and the DataNode Protocol.
- By design, the NameNode never initiates any RPCs. Instead, it **only responds to RPC** requests issued by DataNodes or clients.

# Types of Failures in HDFS(Providing Robustness)

1. **DataNode Failure**

   Data Disk Failure, Heartbeats (Solution- Re-Replication)

   Data Integrity - the Datanode arrived is corrupted(Solution- checksum)

2. **Network Failure**

Cluster Rebalancing-In the event of a sudden high demand for a particular file, a scheme might dynamically create additional replicas and rebalance other data in the cluster.

3. **Namenode Failure**

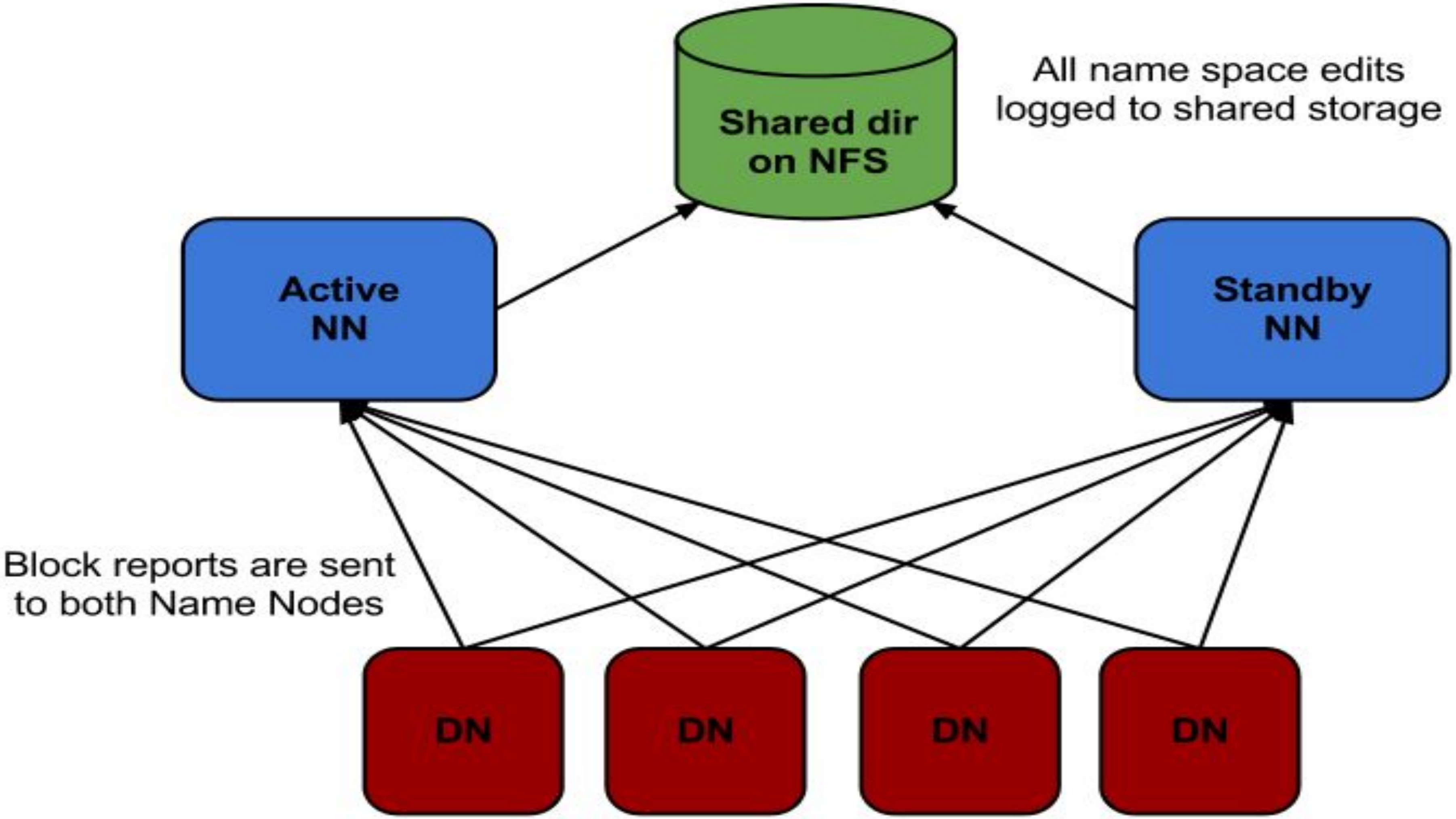
   The FsImage and the EditLog are central data structures of HDFS. A corruption of these files can cause the HDFS instance to be non-functional.

   Solution

   - To synchronous updating of multiple copies of the FsImage and EditLog
   - To enable High Availability using multiple NameNodes

https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

# Multiple Name node(Hadoop 2.x)



Shared dir on NFS

All name space edits logged to shared storage

Active NN

Standby NN

Block reports are sent to both Name Nodes

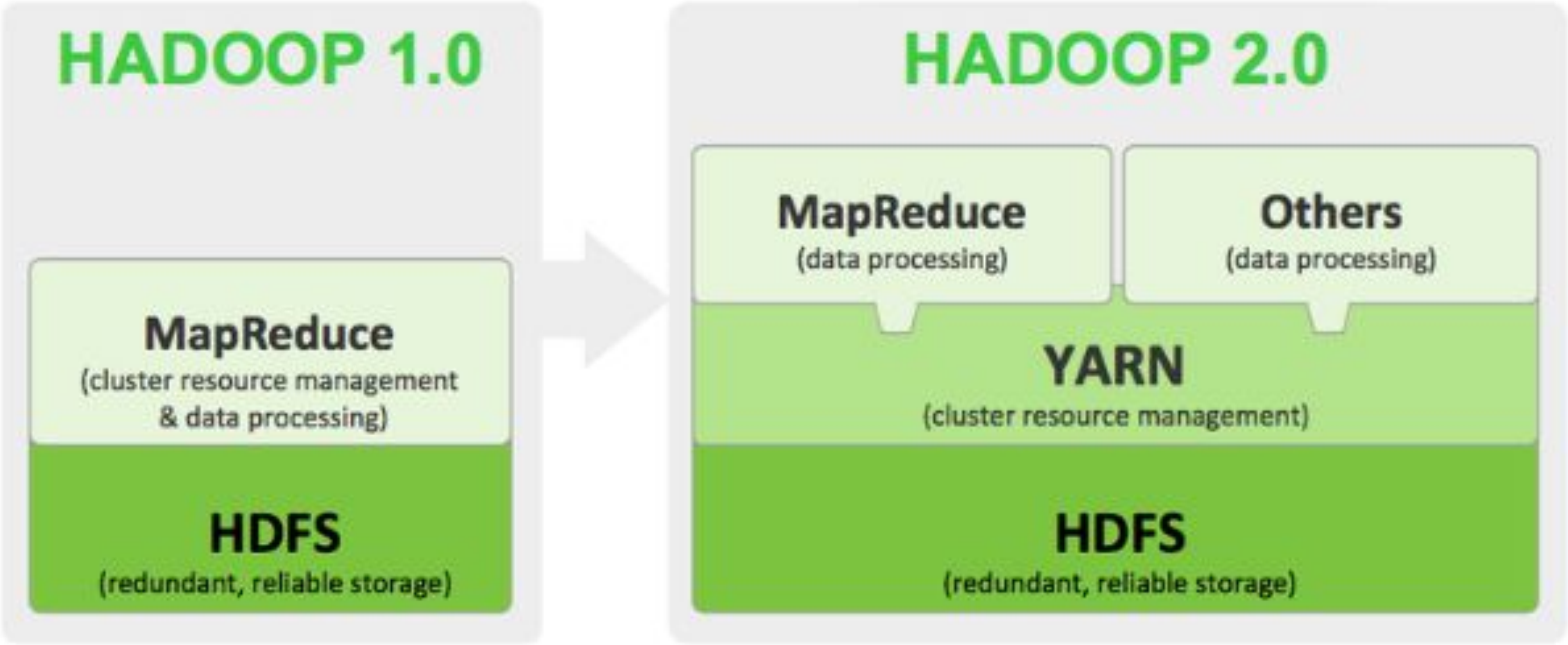
DN

DN

DN

DN

# Hadoop Version



HADOOP 1.0

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

HADOOP 2.0

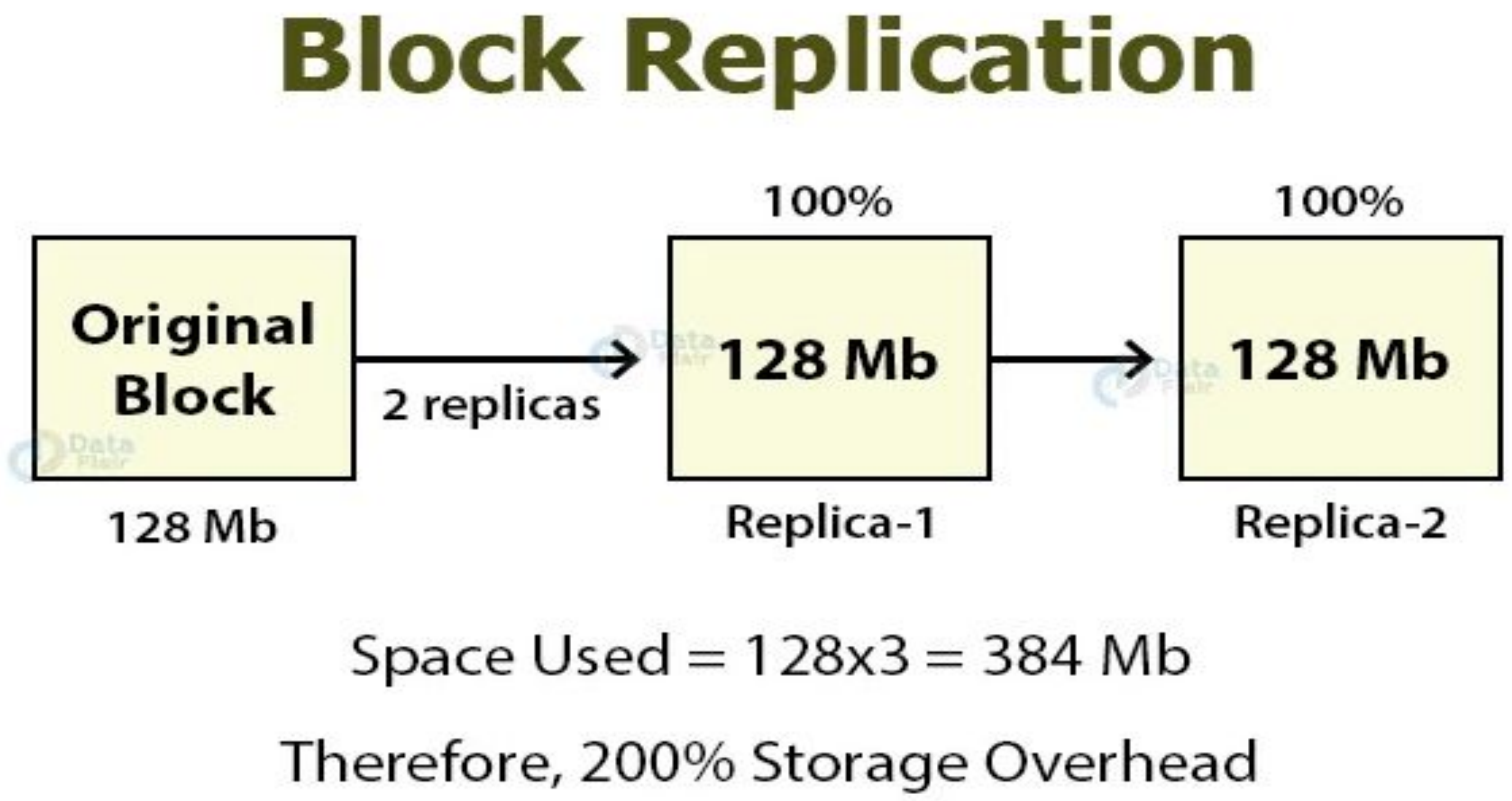
MapReduce
(data processing)

Others
(data processing)

YARN
(cluster resource management)

HDFS
(redundant, reliable storage)

# Erasure Coding(Hadoop 3.x)

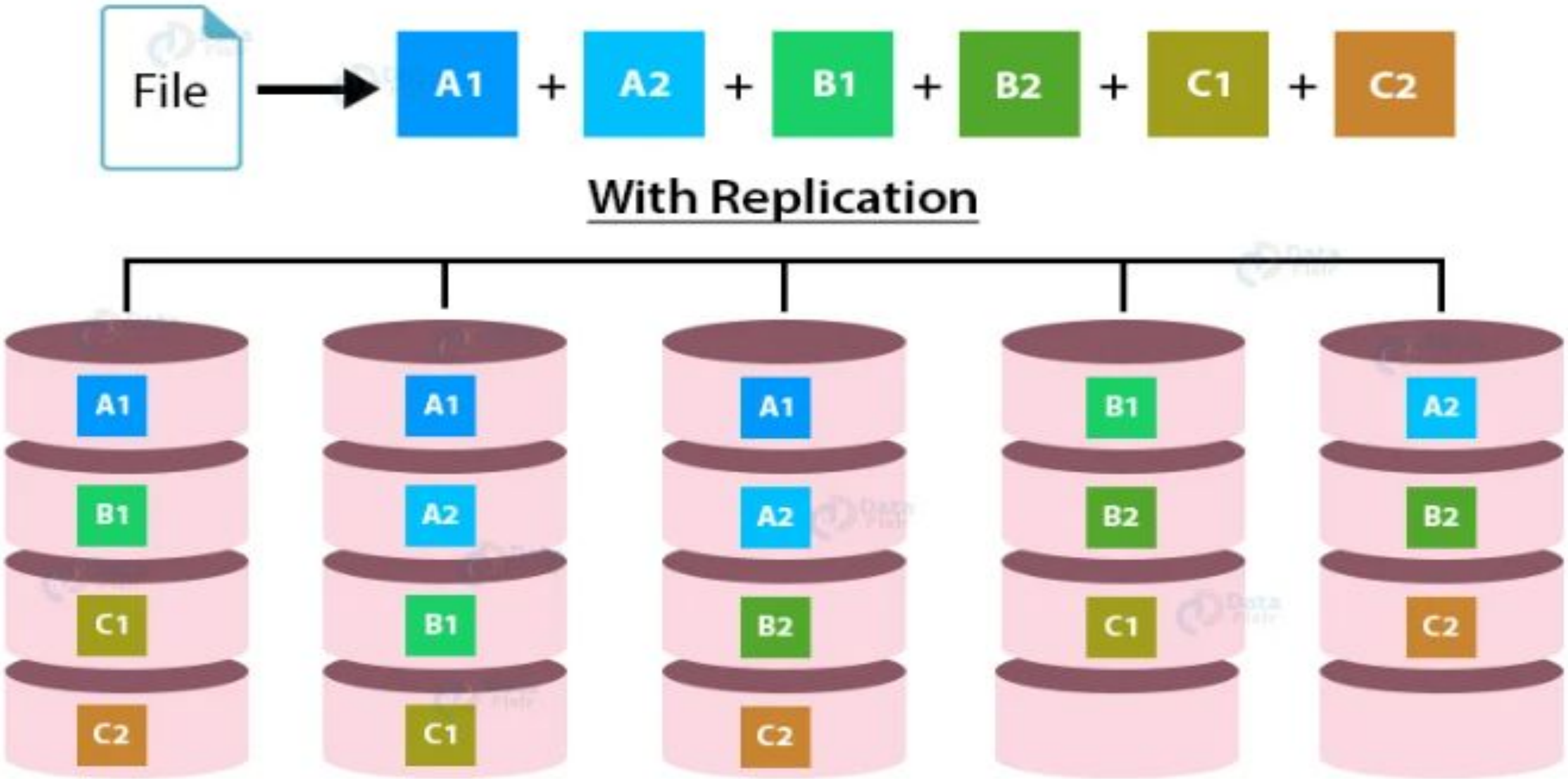


## Block Replication

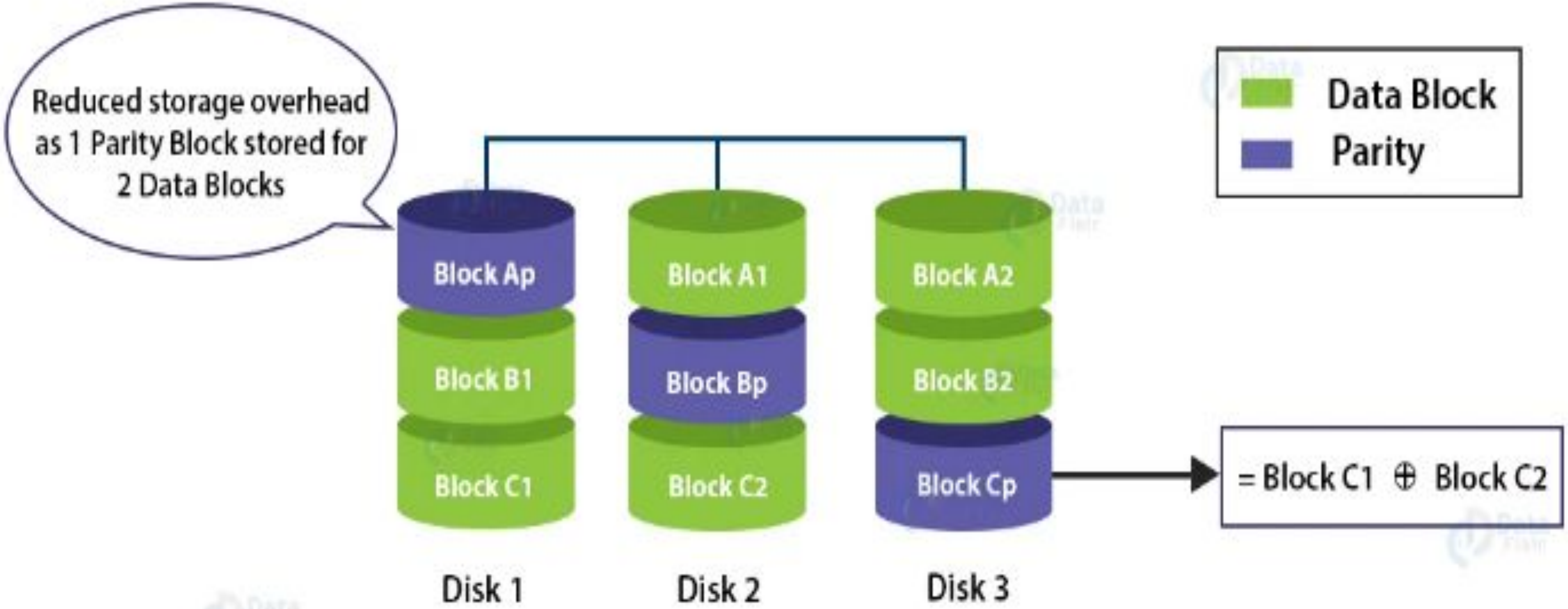
Space Used = 128x3 = 384 Mb

Therefore, 200% Storage Overhead

# 6 blocks with replication factor 3= 18 blocks



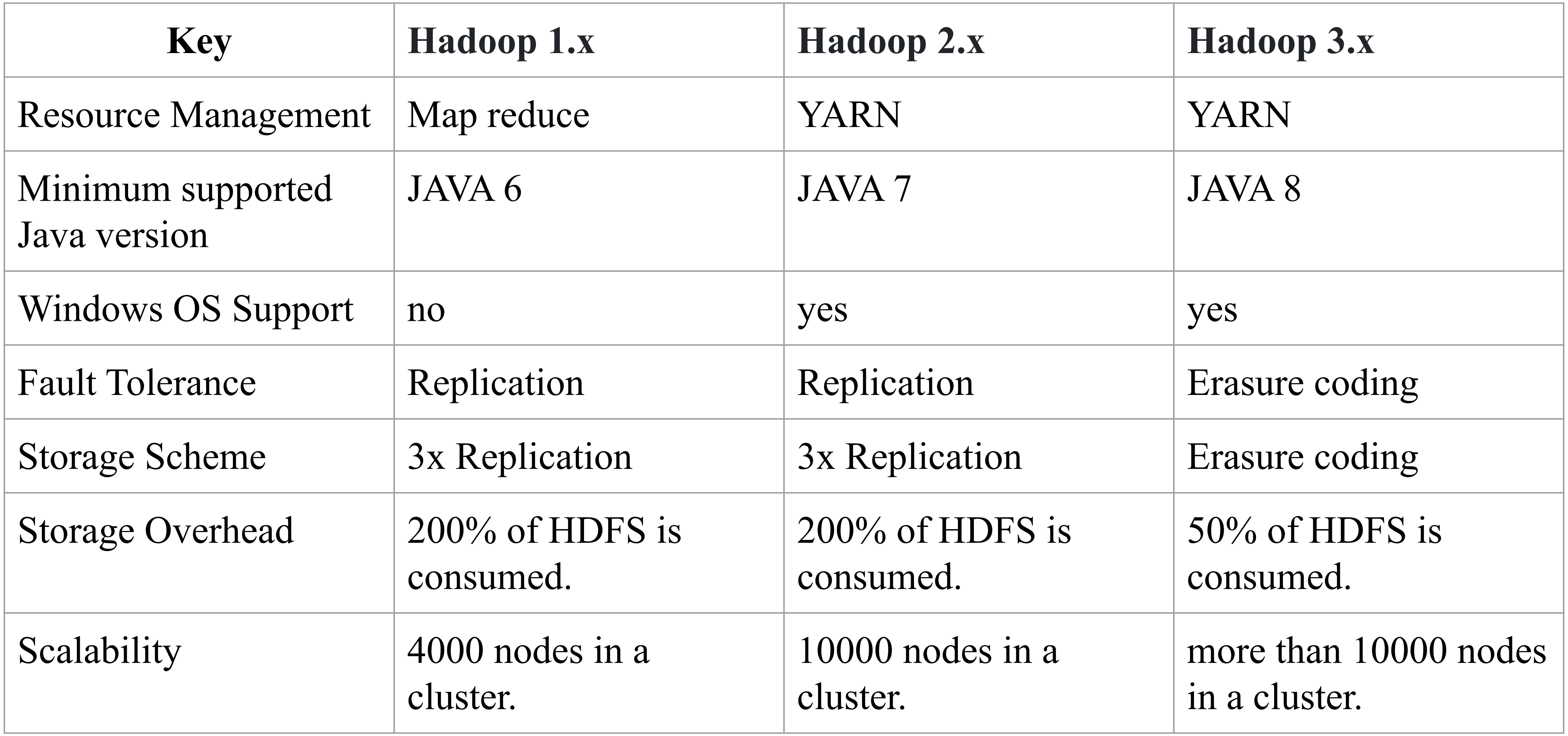
# Erasure Coding



# Erasure Coding(Cont.)

- In storage systems, the **Redundant Array of Inexpensive Disks (RAID)** uses Erasure Coding.
- RAID implements Erasure Coding by **striping**, that is, dividing logically sequential data such as file into smaller units (bit, byte, or block) and storing consecutive units on different disks.
- For each strip of the original dataset, a certain number of parity cells are calculated based on the **Erasure Coding algorithm** and stored, the process known as **encoding**.
- The error in any striping cell can be recovered from the calculation based on the remaining data and parity cells; the process known as **decoding**.
- Thus using Erasure Coding in HDFS improves storage efficiency while providing the same level of fault tolerance and data durability as traditional replication-based HDFS deployment.

# Difference between hadoop 1.x,2.x,3.x



| Key | Hadoop 1.x | Hadoop 2.x | Hadoop 3.x |
|---|---|---|---|
| Resource Management | Map reduce | YARN | YARN |
| Minimum supported Java version | JAVA 6 | JAVA 7 | JAVA 8 |
| Windows OS Support | no | yes | yes |
| Fault Tolerance | Replication | Replication | Erasure coding |
| Storage Scheme | 3x Replication | 3x Replication | Erasure coding |
| Storage Overhead | 200% of HDFS is consumed. | 200% of HDFS is consumed. | 50% of HDFS is consumed. |
| Scalability | 4000 nodes in a cluster. | 10000 nodes in a cluster. | more than 10000 nodes in a cluster. |

# References

- Book
  - Big Data and Analytics – Seema Acharya and Subhashini C – Wiley India
  - https://hadoop.apache.org/docs/r2.10.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html