UNIVERSITAT DE BARCELONA

Master in Fundamental Principles of Data Science

Dr Rohit Kumar

UNIVERSITAT DE BARCELONA

Hadoop

UNIVERSITAT DE BARCELONA

What is Hadoop

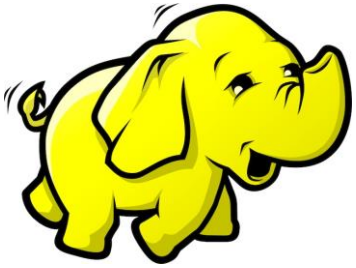
Apache Hadoop is an open source software framework used to develop data processing applications which are executed in a distributed computing environment.

In Hadoop, data resides in a distributed file system which is called as a **Hadoop Distributed File system.**

The processing model is based on **'Data Locality'** concept wherein computational logic is sent to cluster nodes(server) containing data.

UNIVERSITAT DE BARCELONA

Hadoop



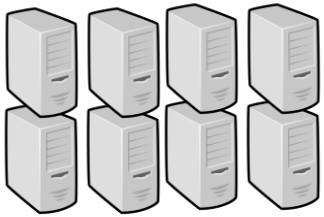
Distributed Computation

A new programing model called ***MapReduce***
Is used to run computation.

Distributed Storage

HDFS a distributed file system provides storage

Server cluster



Hadoop can run on commodity hardware

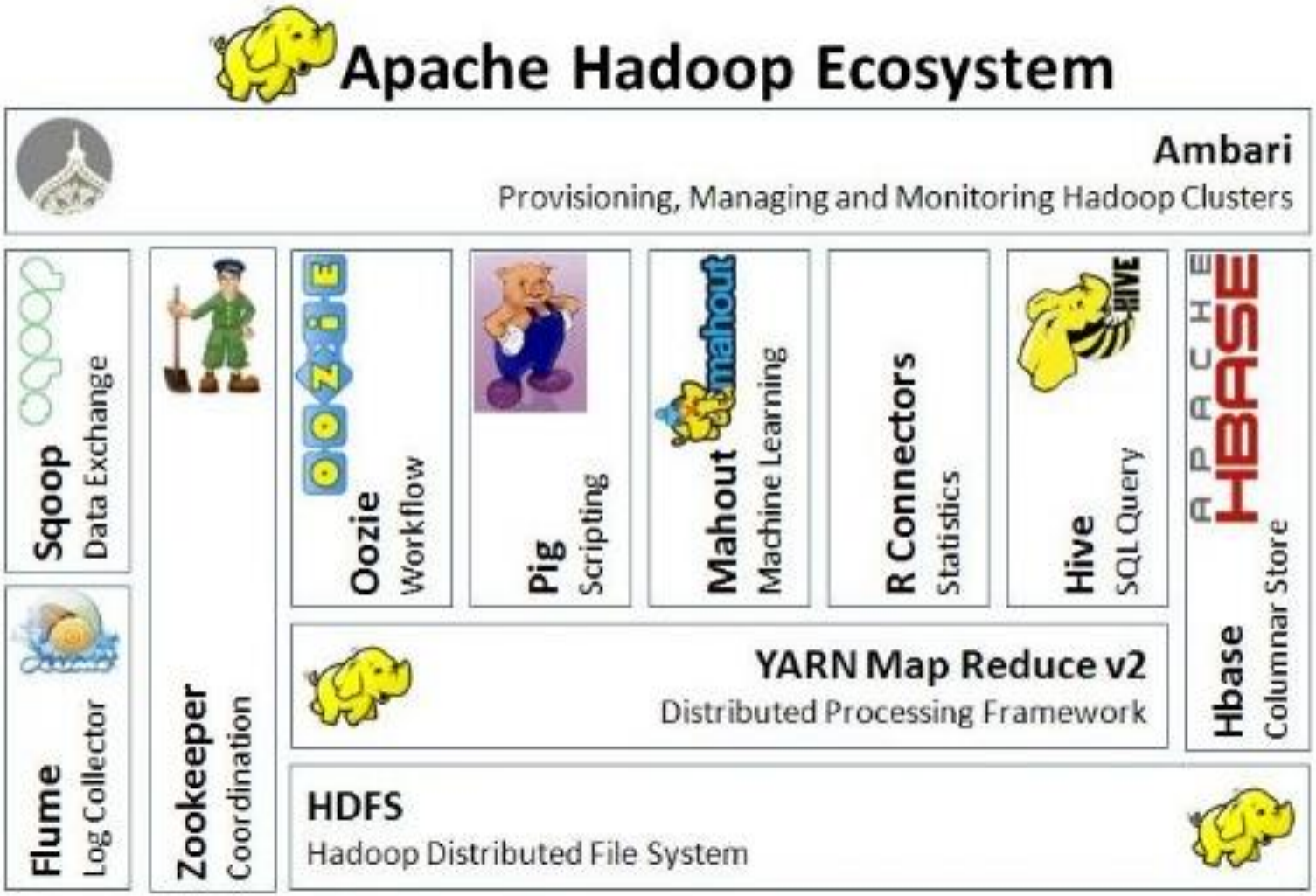
UNIVERSITAT DE BARCELONA

Hadoop(Continued..)

- Hadoop is a platform that provides both distributed storage and computational capabilities.
- It was built mainly as a ***batch processing*** system.
- It consist of one cycle of loading data from HDFS applying Map job then running the reduce job and finally writing the output back into HDFS.
- In case of ***iterative*** computation the data needs to be loaded again!

UNIVERSITAT DE BARCELONA

Hadoop EcoSystem



UNIVERSITAT DE BARCELONA

HDFS

UNIVERSITAT DE BARCELONA

Some Use case

- In 2010, **Facebook** claimed to have one of the largest HDFS cluster storing **21 Petabytes** of data.
- In 2012, **Facebook** declared that they have the largest single HDFS cluster with more than **100 PB** of data**.**
- And **Yahoo**! has more than **100,000 CPU** in over **40,000 servers** running Hadoop, with its biggest Hadoop cluster running **4,500 nodes**. All told, Yahoo! stores **455 petabytes** of data in HDFS.

UNIVERSITAT DE BARCELONA

HDFS?

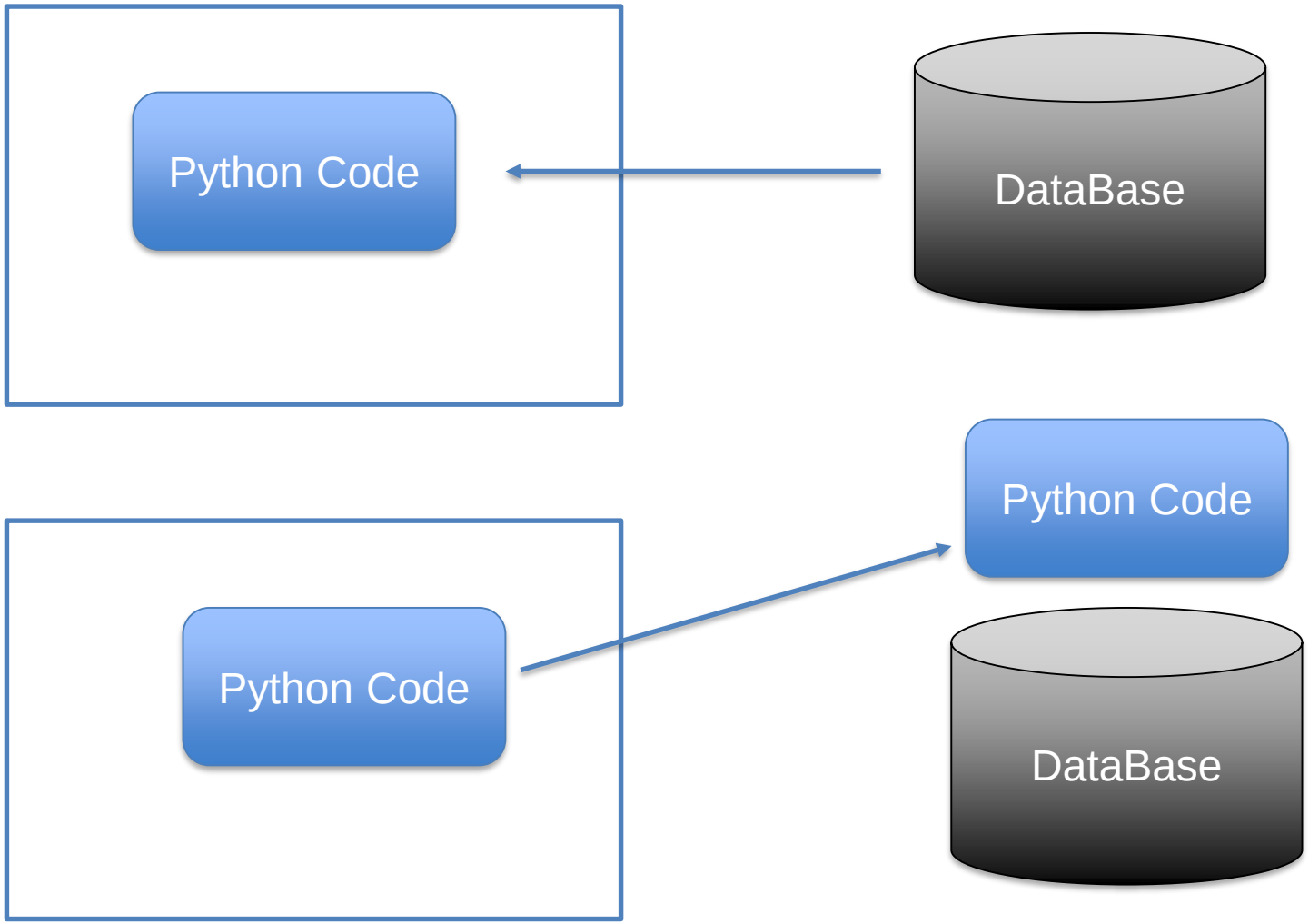
Hadoop Distributed file system or HDFS is a Java based distributed file system that allows you to store large data across multiple nodes in a Hadoop cluster.

It is based out of commodity hardware.

One of the main advantage is **Data Locality**

UNIVERSITAT DE BARCELONA

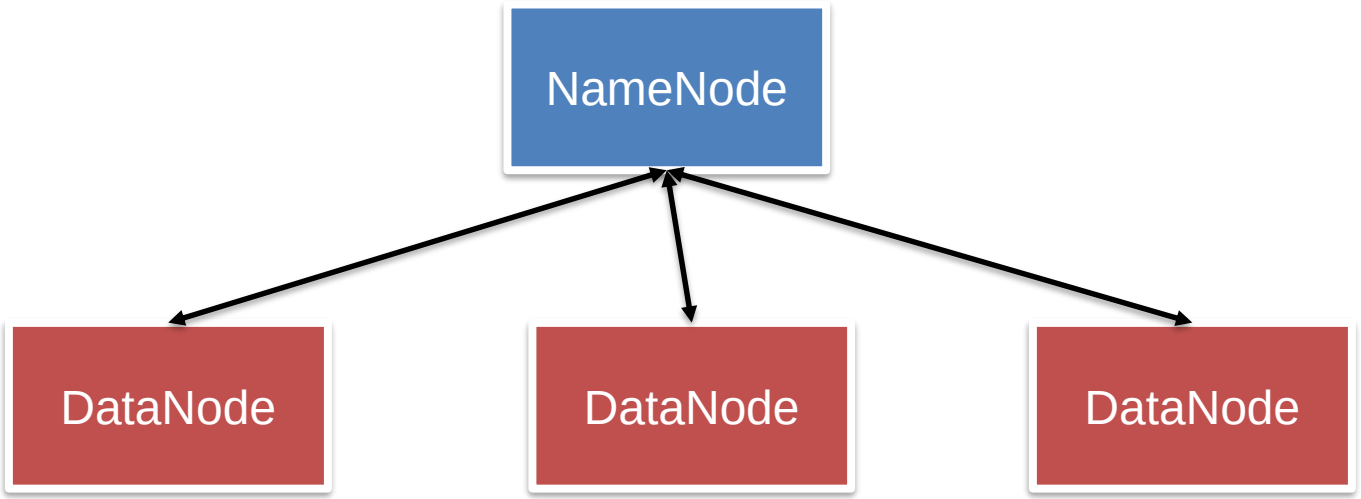
Data Locality



UNIVERSITAT DE BARCELONA

HDFS Architecture

- **Apache HDFS** is a block-structured file system where each file is divided into blocks of a pre-determined size. These blocks are stored across a cluster of one or several machines. Apache Hadoop HDFS Architecture follows a *Master/Slave Architecture*, where a cluster comprises of a single NameNode (Master node) and all the other nodes are DataNodes (Slave nodes).



UNIVERSITAT DE BARCELONA

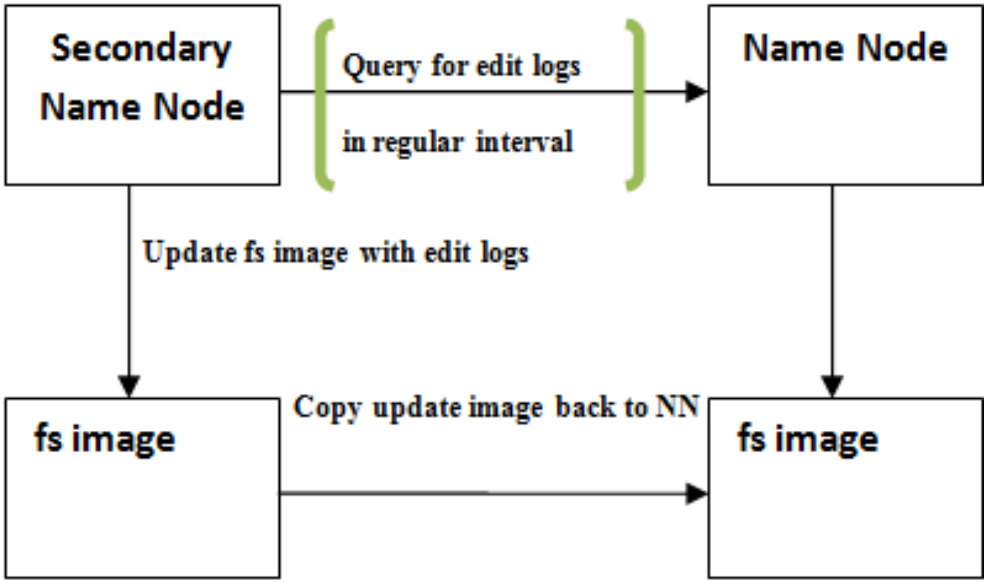
HDFS Architecture

- **NameNode:** NameNode can be considered as a master of the system. It maintains the file system tree and the metadata for all the files and directories present in the system. Two files **'Namespace image'** and the **'edit log'** are used to store metadata information. Namenode has knowledge of all the datanodes containing data blocks for a given file, however, it does not store block locations persistently. This information is reconstructed every time from datanodes when the system starts.
- **DataNode:** DataNodes are slaves which reside on each machine in a cluster and provide the actual storage. It is responsible for serving, read and write requests for the clients.

UNIVERSITAT DE BARCELONA

Secondary NameNode

The Secondary NameNode works concurrently with the primary NameNode as a **helper daemon. It is not a backup of NameNode but just a helper.**

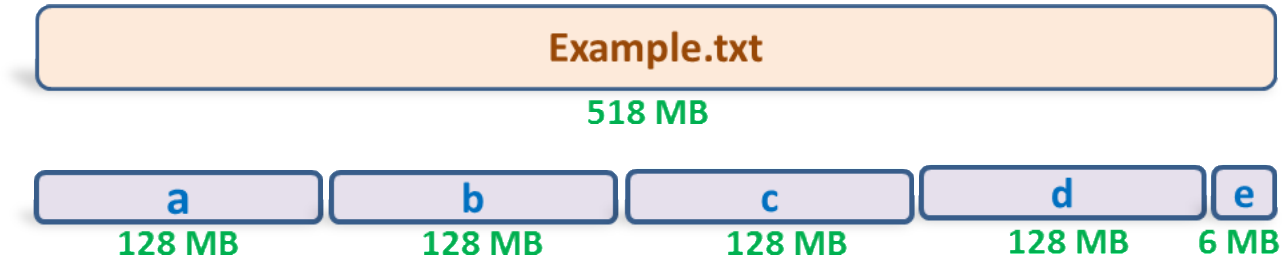


Secondary NameNode performs regular checkpoints in HDFS. Therefore, it is also called CheckpointNode.

UNIVERSITAT DE BARCELONA

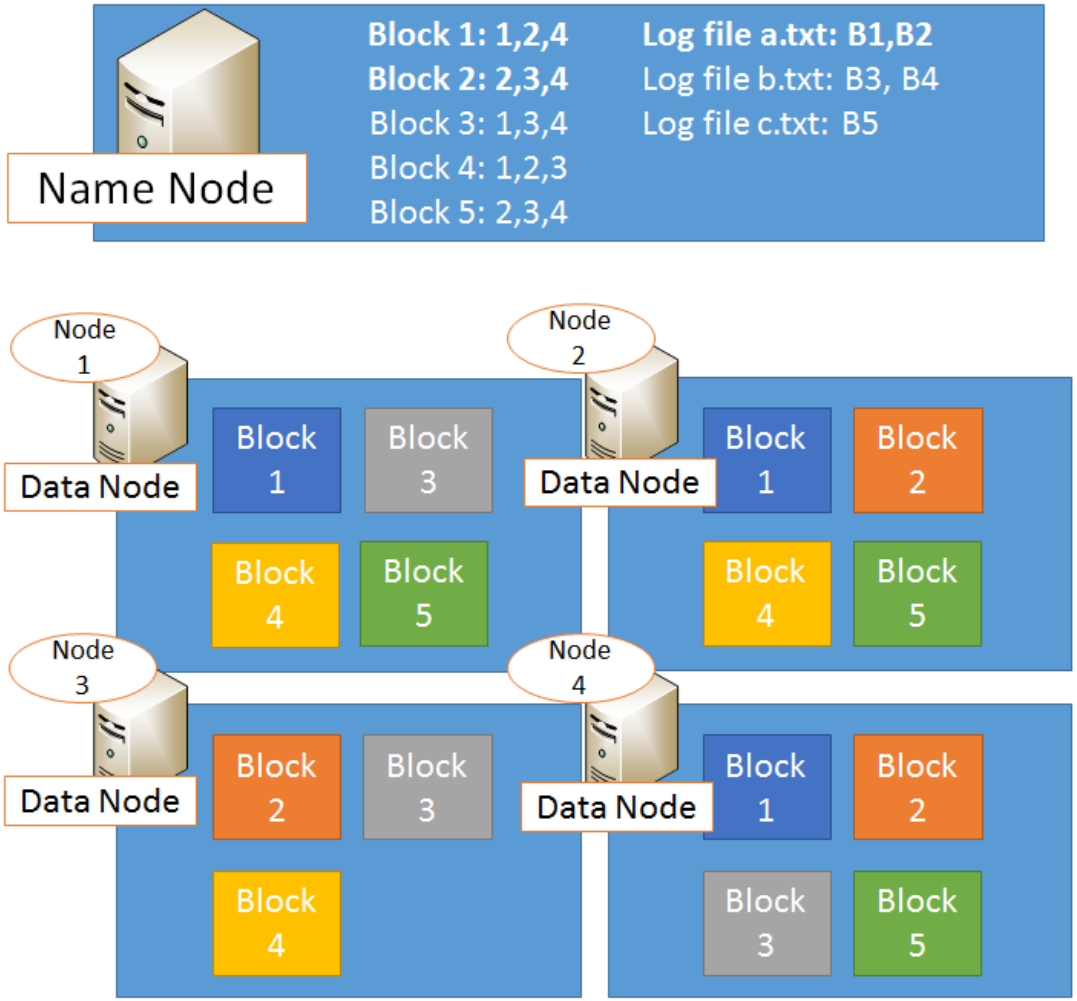
Blocks

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 128 MB in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x) which you can configure as per your requirement.



UNIVERSITAT DE BARCELONA

Replication management



Replication Factor 3

UNIVERSITAT DE BARCELONA

S3

UNIVERSITAT DE BARCELONA

What is object storage?

Object storage (also referred to as object-based storage) is a general term that refers to the way in which we organize and work with units of storage, called objects. Every object contains three things:

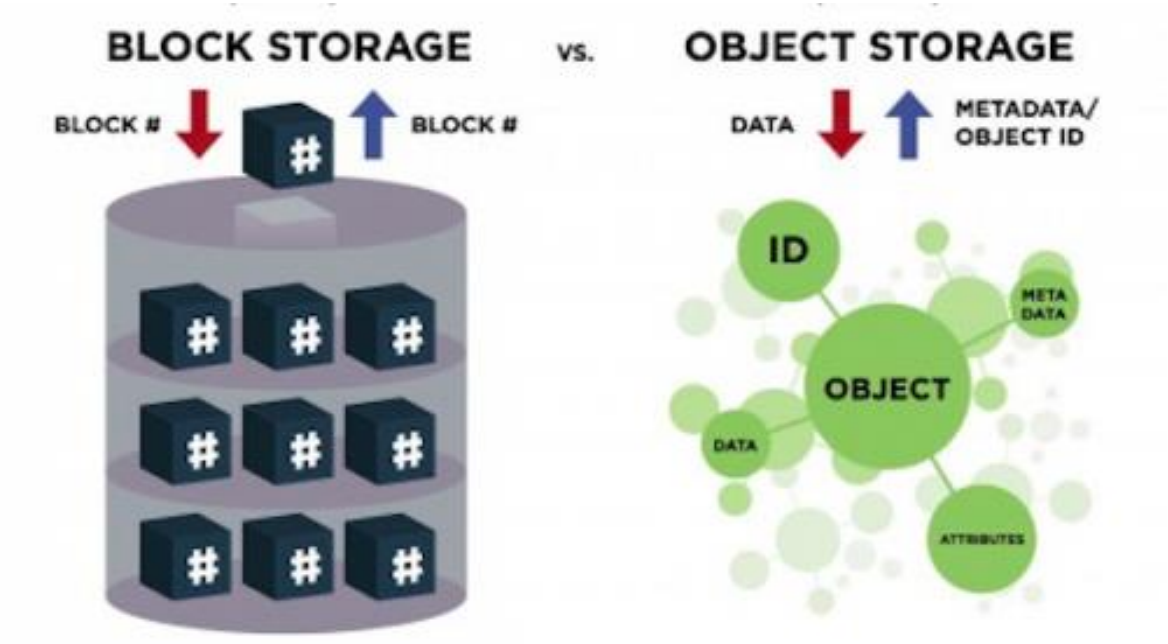
- The data itself.
- An expandable amount of metadata.
- A globally unique identifier.

How block storage and object storage differ

With block storage, files are split into evenly sized blocks of data, each with its own address but with no additional information (metadata) to provide more context for what that block of data is.

Object storage, by contrast, doesn’t split files up into raw blocks of data. Instead, entire clumps of data are stored in, yes, an object that contains the data, metadata, and the unique identifier.

How block storage and object storage differ



UNIVERSITAT DE BARCELONA

Object Storage vs Block Storage

	OBJECT STORAGE	BLOCK STORAGE
PERFORMANCE	Performs best for big content and high stream throughput	Performs best for smaller files
GEOGRAPHY	Data can be stored across multiple regions	Data typically needs to be shared locally
SCALABILITY	Scales infinitely to petabytes and beyond	Potentially scales up to millions of files, but can't handle more
ANALYTICS	Customizable metadata, not limited to number of tags	Limited number of set metadata tags

UNIVERSITAT DE BARCELONA

When to use what?

Object storage generally doesn’t provide you with the ability to incrementally edit one part of a file (as block storage does). Objects have to be manipulated as a whole unit, requiring the entire object to be accessed, updated, then re-written in their entirety. That can have performance implications.

So if you need to store data for distributed processing use HDFS but if you need a highly elastic cheap space for fast data storage use object store.

UNIVERSITAT DE BARCELONA

Amazon S3

Why it is called S3?

It is an object store

Simple Storage Service

Bucket and Objects

UNIVERSITAT DE BARCELONA

Amazon S3

Amazon Simple Storage Service (S3) is a storage for the internet. It is designed for large-capacity, low-cost storage provision across multiple geographical regions. Amazon S3 provides developers and IT teams
with **Secure**, **Durable** and **Highly Scalable** object storage.

UNIVERSITAT DE BARCELONA

S3 is Secure

- Encryption to the data that you store. It can happen in two ways:
  - Client Side Encryption
  - Server Side Encryption
- Multiple copies are maintained to enable regeneration of data in case of data corruption
- *Versioning,* wherein each edit is archived for a potential retrieval.

UNIVERSITAT DE BARCELONA

S3 is **Durable**

- It regularly verifies the integrity of data stored using checksums e.g. if S3 detects there is any corruption in data, it is immediately repaired with the help of replicated data.
- Even while storing or retrieving data, it checks incoming network traffic for any corrupted data packets.

S3 is **Highly Scalable**, since it automatically scales your storage according to your requirement and you only pay for the storage you use.

UNIVERSITAT DE BARCELONA

Storage Class

When we talk about data, it can be of two types-

- Data which is to be accessed frequently.
- Data which is accessed not that frequently.

Amazon S3 offers a range of storage classes designed for different use cases.

UNIVERSITAT DE BARCELONA

Storage Class

- **S3 Standard** for general-purpose storage of frequently accessed data
- **S3 Intelligent-Tiering** for data with unknown or changing access patterns;
- **S3 Standard-Infrequent Access (S3 Standard-IA)** and **S3 One Zone-Infrequent Access (S3 One Zone-IA)** for long-lived, but less frequently accessed data
- **Amazon S3 Glacier (S3 Glacier)** and **Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive)** for long-term archive and digital preservation.

UNIVERSITAT DE BARCELONA

Storage Class

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	millseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

UNIVERSITAT DE BARCELONA

Hadoop Processing

UNIVERSITAT DE BARCELONA

Lets see how distributed computing work

Counting words in a document (*You might ask why we would do such thing ☺!! But it's a key step in Document search and matching*)

UNIVERSITAT DE BARCELONA

One simple approach

Use a Hash Table

“Who is Ram

Ram is who

Ram is king

King of Ajodhya”

Key	Value

UNIVERSITAT DE BARCELONA

One simple approach

Use a Hash Table

"Who is Ram

Ram is who

Ram is king

King of Ajodhya"

Key	Value
Who	1

UNIVERSITAT DE BARCELONA

One simple approach

Use a Hash Table

“Who is Ram

Ram is who

Ram is king

King of Ajodhya”

Key	Value
Who	1
Is	1

UNIVERSITAT DE BARCELONA

One simple approach

Use a Hash Table

“Who is Ram

Ram is who

Ram is king

King of Ajodhya”

Key	Value
Who	1
Is	1
Ram	1

UNIVERSITAT DE BARCELONA

One simple approach

Use a Hash Table

“Who is Ram

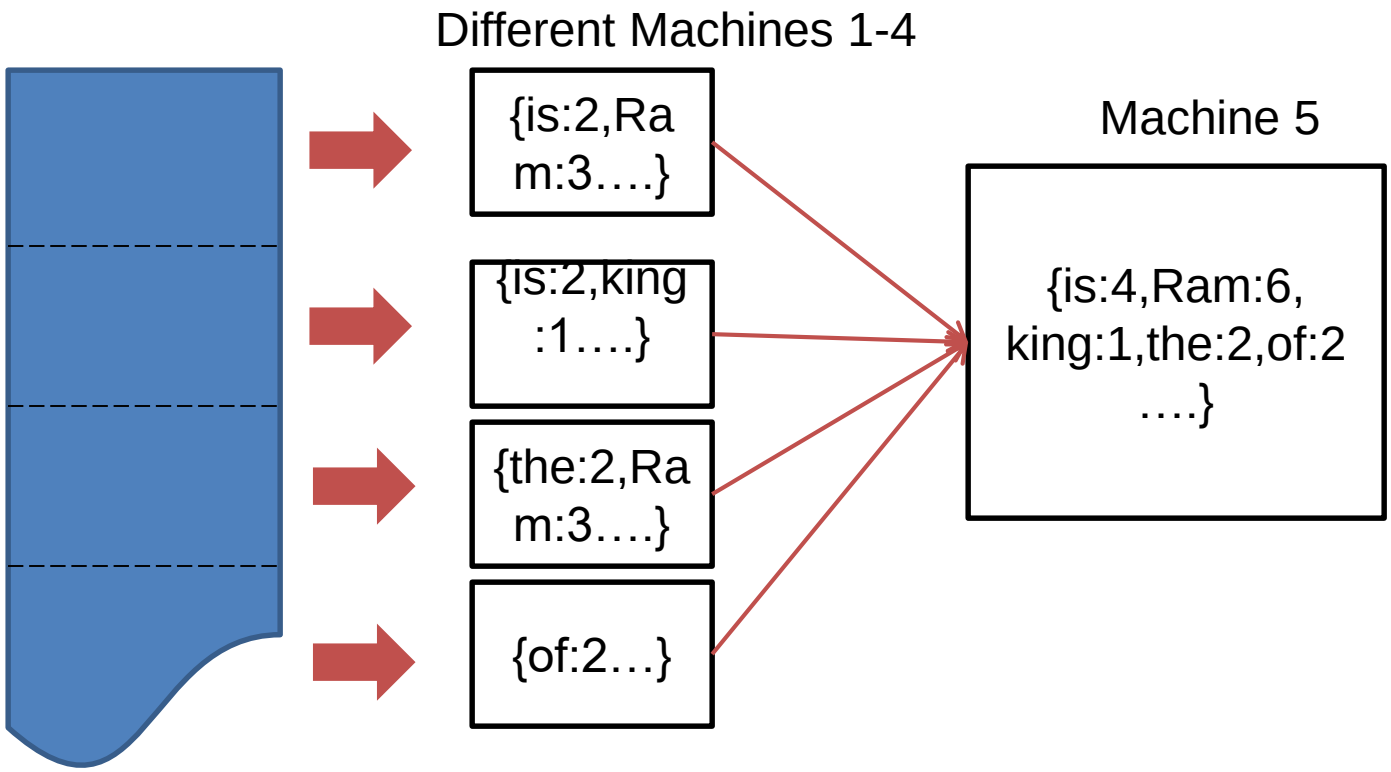
Ram is who

Ram is king

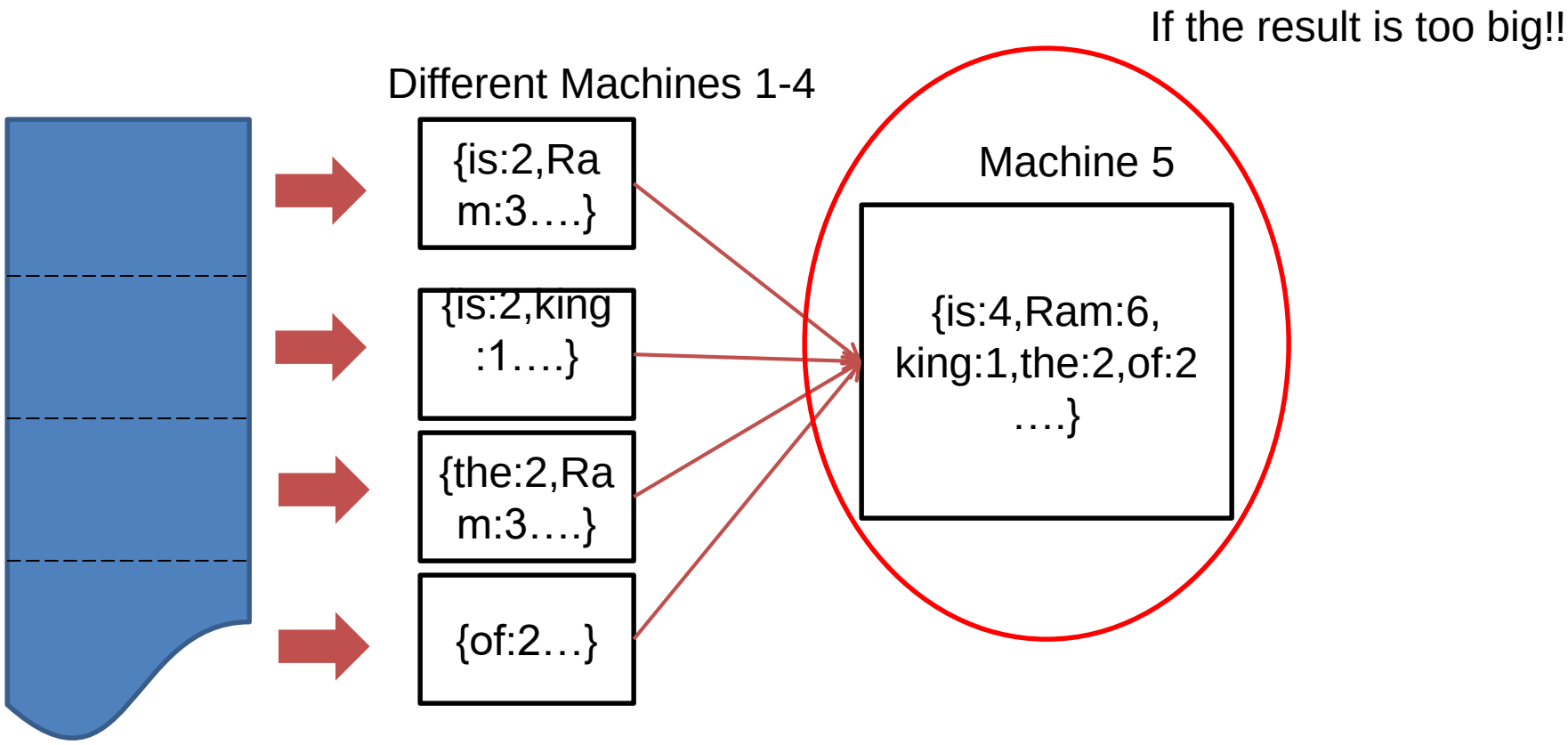
King of Ajodhya”

Key	Value
Who	1
Is	1
Ram	2

What if the document is very big?

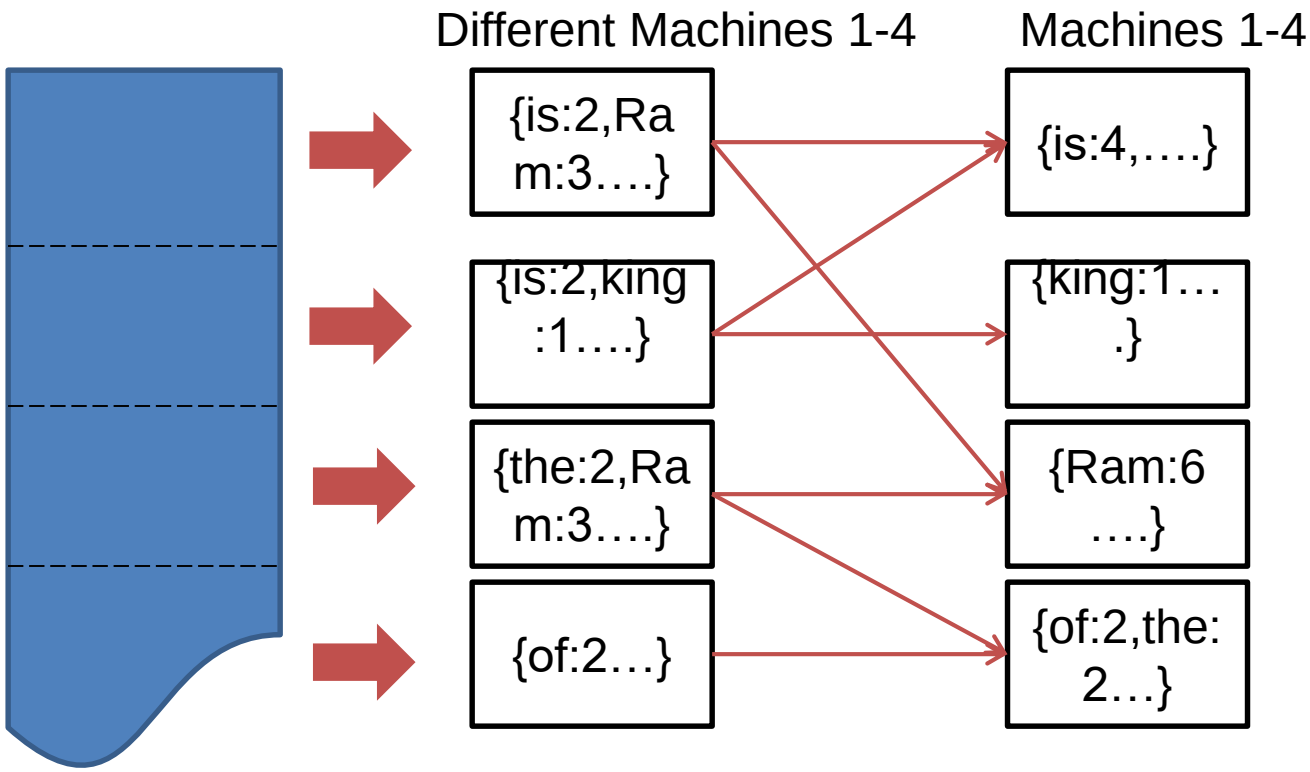


What if the document is very big?



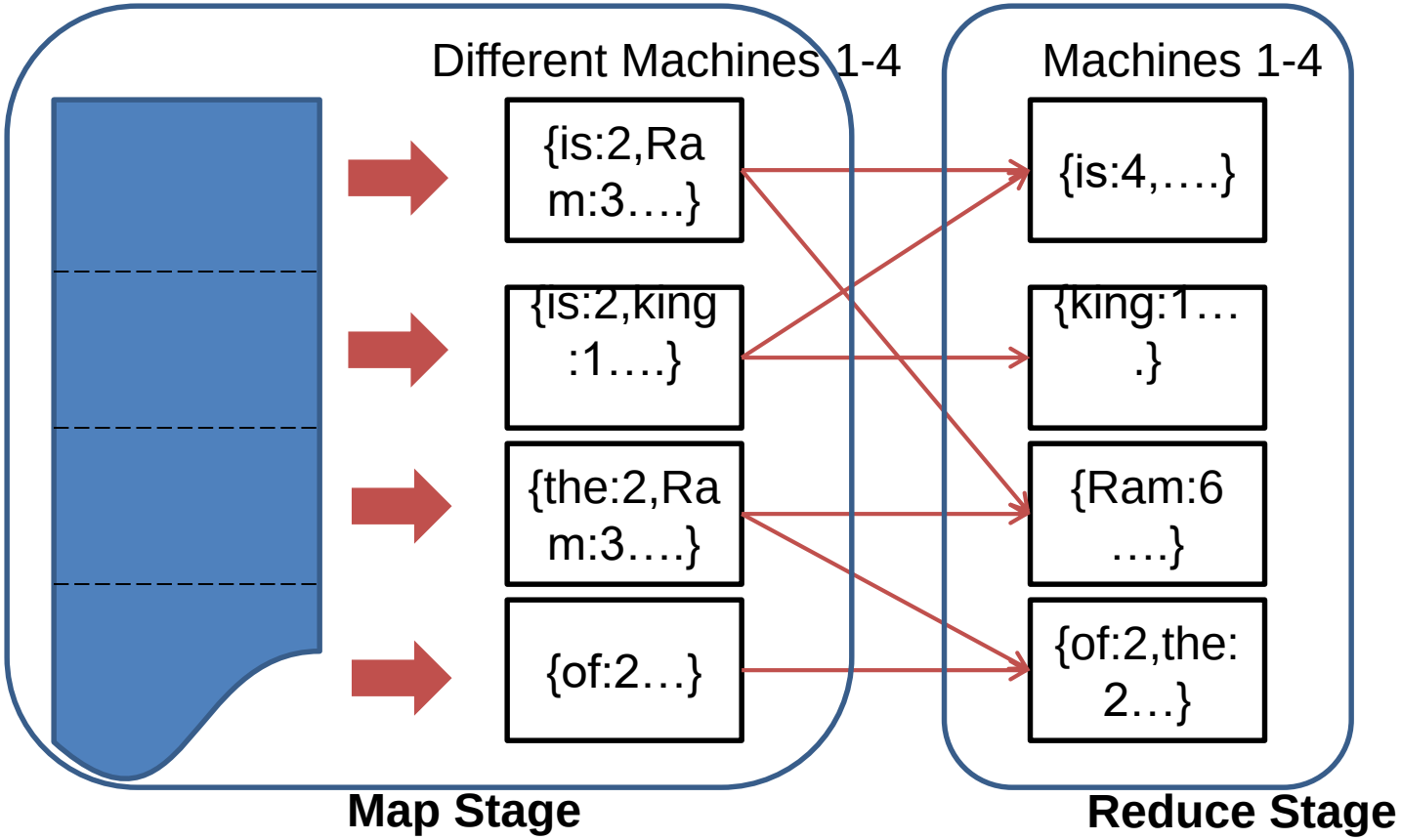
UNIVERSITAT DE BARCELONA

Map Reduce



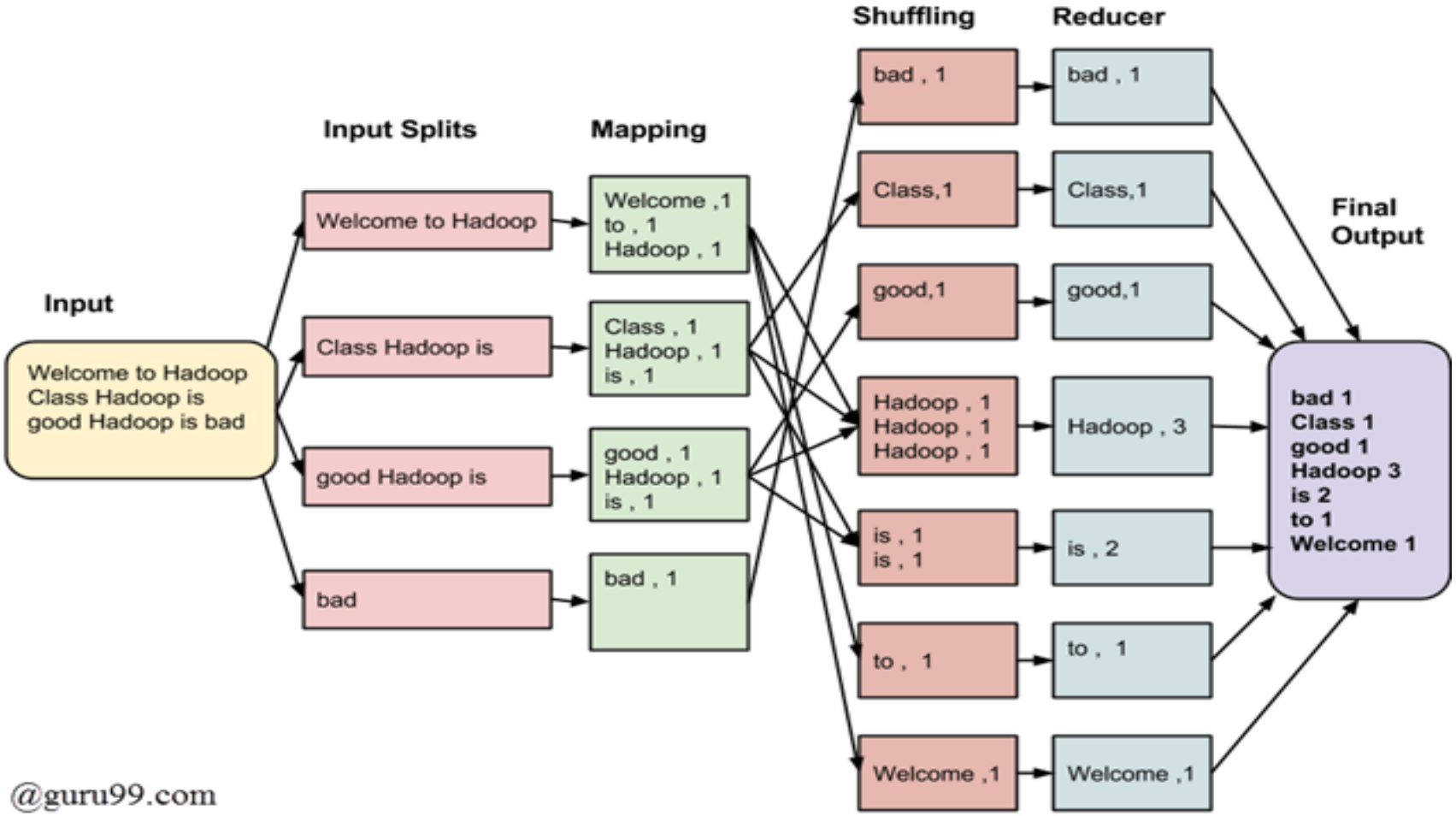
UNIVERSITAT DE BARCELONA

Map Reduce



UNIVERSITAT DE BARCELONA

Different Phases in MapReduce



https://www.guru99.com/introduction-to-mapreduce.html

UNIVERSITAT DE BARCELONA

MapReduce Phases

- **Input Splits:** An input to a MapReduce job is divided into fixed-size pieces called **input splits** Input split is a chunk of the input that is consumed by a single map
- **Mapping:** This is the very first phase in the execution of map-reduce program. In this phase data in each split is passed to a mapping function to produce output values. In our example, a job of mapping phase is to count a number of occurrences of each word from input splits (more details about input-split is given below) and prepare a list in the form of <word, frequency>

UNIVERSITAT DE BARCELONA

MapReduce Phases

- **Shuffling:** This phase consumes the output of Mapping phase. Its task is to consolidate the relevant records from Mapping phase output. In our example, the same words are clubed together along with their respective frequency.
- **Reducing:** In this phase, output values from the Shuffling phase are aggregated. This phase combines values from Shuffling phase and returns a single output value. In short, this phase summarizes the complete dataset.

UNIVERSITAT DE BARCELONA

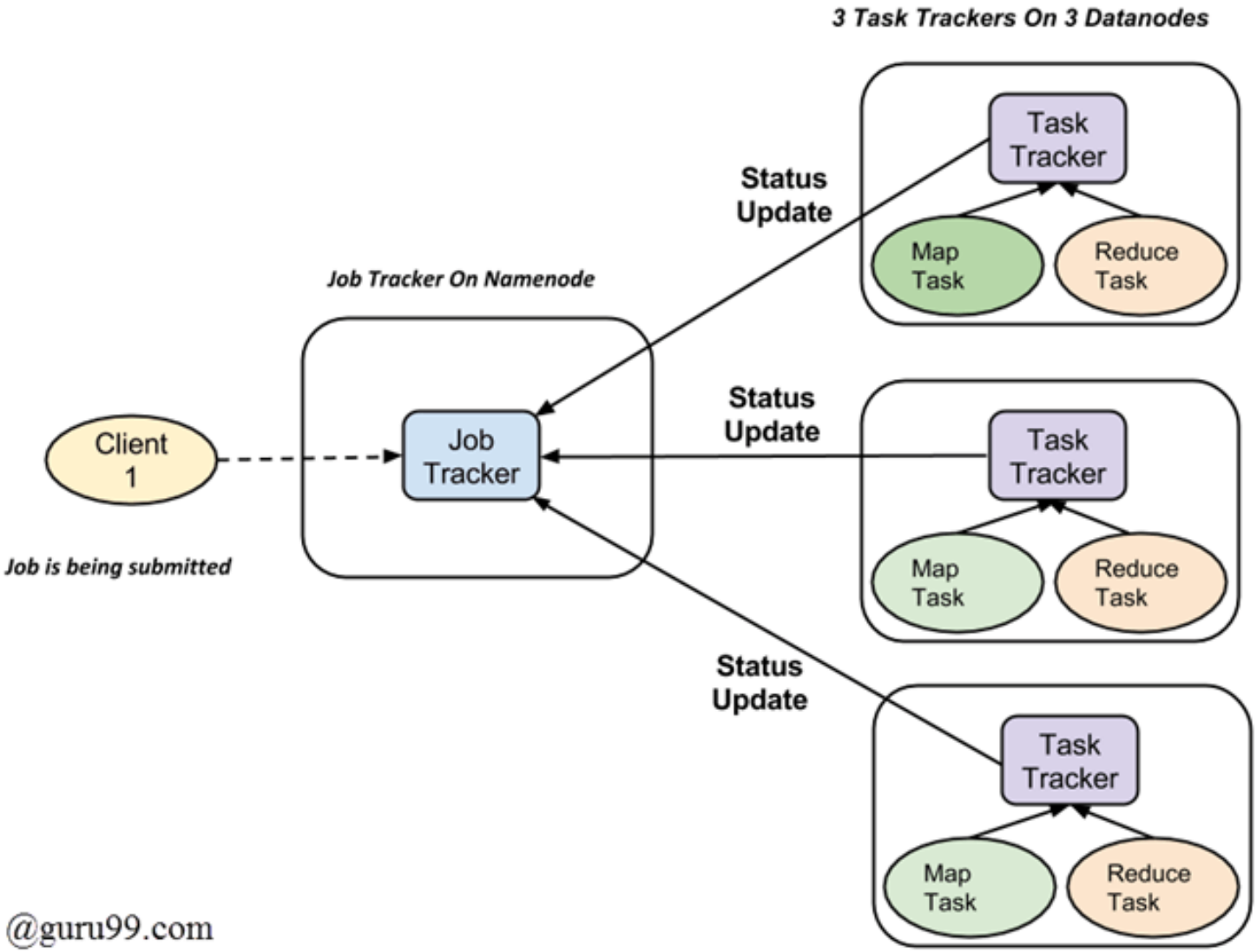
MapReduce Work Organization

The complete execution process (execution of Map and Reduce tasks, both) is controlled by two types of entities called a

- **Jobtracker**: Acts like a **master** (responsible for complete execution of submitted job)
- **Multiple Task Trackers**: Acts like **slaves,** each of them performing the job

UNIVERSITAT DE BARCELONA

MapReduce Work Organization



UNIVERSITAT DE BARCELONA

Hadoop 2.0

Hadoop v1.0

MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage

2006-2012

Hadoop v2.0

MapReduce

Other Data Processing Frameworks

YARN
Resource Management

HDFS
Distributed File Storage

2013

UNIVERSITAT DE BARCELONA

YARN

- YARN enabled the users to perform operations as per requirement by using a variety of tools like **Spark** for real-time processing, **Hive** for SQL, **HBase** for NoSQL and others.
- Apart from Resource Management, YARN also performs Job Scheduling. YARN performs all your processing activities by allocating resources and scheduling tasks.

UNIVERSITAT DE BARCELONA

YARN Architecture

- **Resource Manager:** Runs on a master daemon and manages the resource allocation in the cluster.
- **Node Manager:** They run on the slave daemons and are responsible for the execution of a task on every single Data Node.
- **Application Master:** Manages the user job lifecycle and resource needs of individual applications. It works along with the Node Manager and monitors the execution of tasks.
- **Container:** Package of resources including RAM, CPU, Network, HDD etc on a single node.

UNIVERSITAT DE BARCELONA

Resource Manager

- It is the ultimate authority in resource allocation.
- On receiving the processing requests, it passes parts of requests to corresponding node managers accordingly, where the actual processing takes place.
- It is the arbitrator of the cluster resources and decides the allocation of the available resources for competing applications.
- Optimizes the cluster utilization like keeping all resources in use all the time against various constraints such as capacity guarantees, fairness, and SLAs.
- It has two major components:
  a) Scheduler
  b) Application Manager

UNIVERSITAT DE BARCELONA

Resource manager- *Scheduler*

- The scheduler is responsible for allocating resources to the various running applications subject to constraints of capacities, queues etc.
- It is called a pure scheduler in ResourceManager, which means that it does not perform any monitoring or tracking of status for the applications.
- If there is an application failure or hardware failure, the Scheduler does not guarantee to restart the failed tasks.
- Performs scheduling based on the resource requirements of the applications.
- It has a pluggable policy plug-in, which is responsible for partitioning the cluster resources among the various applications. There are two such plug-ins: **Capacity Scheduler** and **Fair Scheduler**, which are currently used as Schedulers in ResourceManager.

UNIVERSITAT DE BARCELONA

Resource manager- *Application Manager*

- It is responsible for accepting job submissions.
- Negotiates the first container from the Resource Manager for executing the application specific Application Master.
- Manages running the Application Masters in a cluster and provides service for restarting the Application Master container on failure.

UNIVERSITAT DE BARCELONA

Node Manager

- It takes care of individual nodes in a Hadoop cluster and manages user jobs and workflow on the given node.
- It registers with the Resource Manager and sends heartbeats with the health status of the node.
- Its primary goal is to manage application containers assigned to it by the resource manager.
- It keeps up-to-date with the Resource Manager.
- Application Master requests the assigned container from the Node Manager by sending it a Container Launch Context(CLC) which includes everything the application needs in order to run. The Node Manager creates the requested container process and starts it.
- Monitors resource usage (memory, CPU) of individual containers.
- Performs Log management.
- It also kills the container as directed by the Resource Manager

UNIVERSITAT DE BARCELONA

Application Master

- An application is a single job submitted to the framework. Each such application has a unique Application Master associated with it which is a framework specific entity.
- It is the process that coordinates an application's execution in the cluster and also manages faults.
- Its task is to negotiate resources from the Resource Manager and work with the Node Manager to execute and monitor the component tasks.
- It is responsible for negotiating appropriate resource containers from the ResourceManager, tracking their status and monitoring progress.
- Once started, it periodically sends heartbeats to the Resource Manager to affirm its health and to update the record of its resource demands.

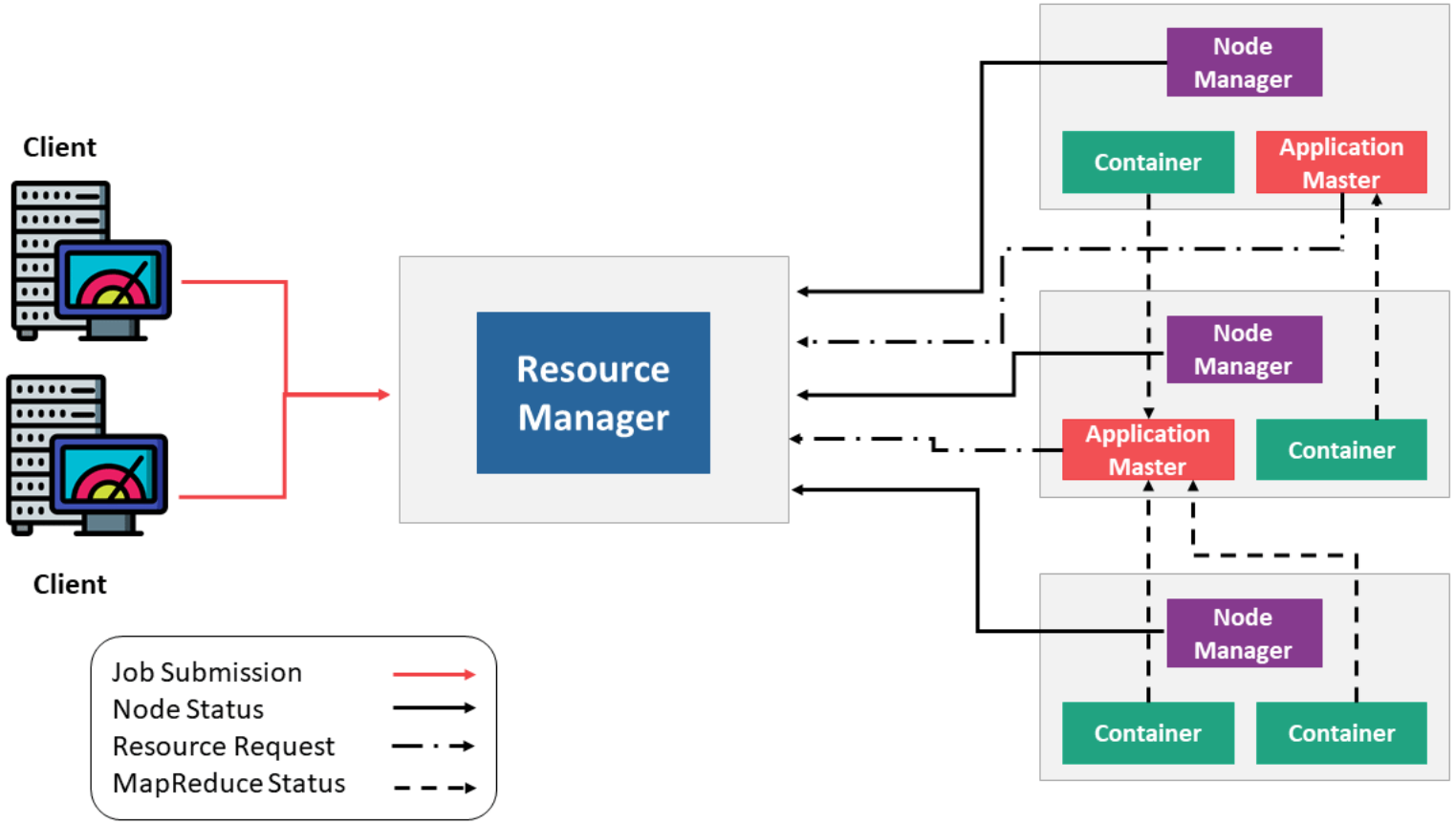
UNIVERSITAT DE BARCELONA

Container

- It is a collection of physical resources such as RAM, CPU cores, and disks on a single node.
- YARN containers are managed by a container launch context which is container life-cycle(CLC). This record contains a map of environment variables, dependencies stored in a remotely accessible storage, security tokens, payload for Node Manager services and the command necessary to create the process.
- It grants rights to an application to use a specific amount of resources (memory, CPU etc.) on a specific host.

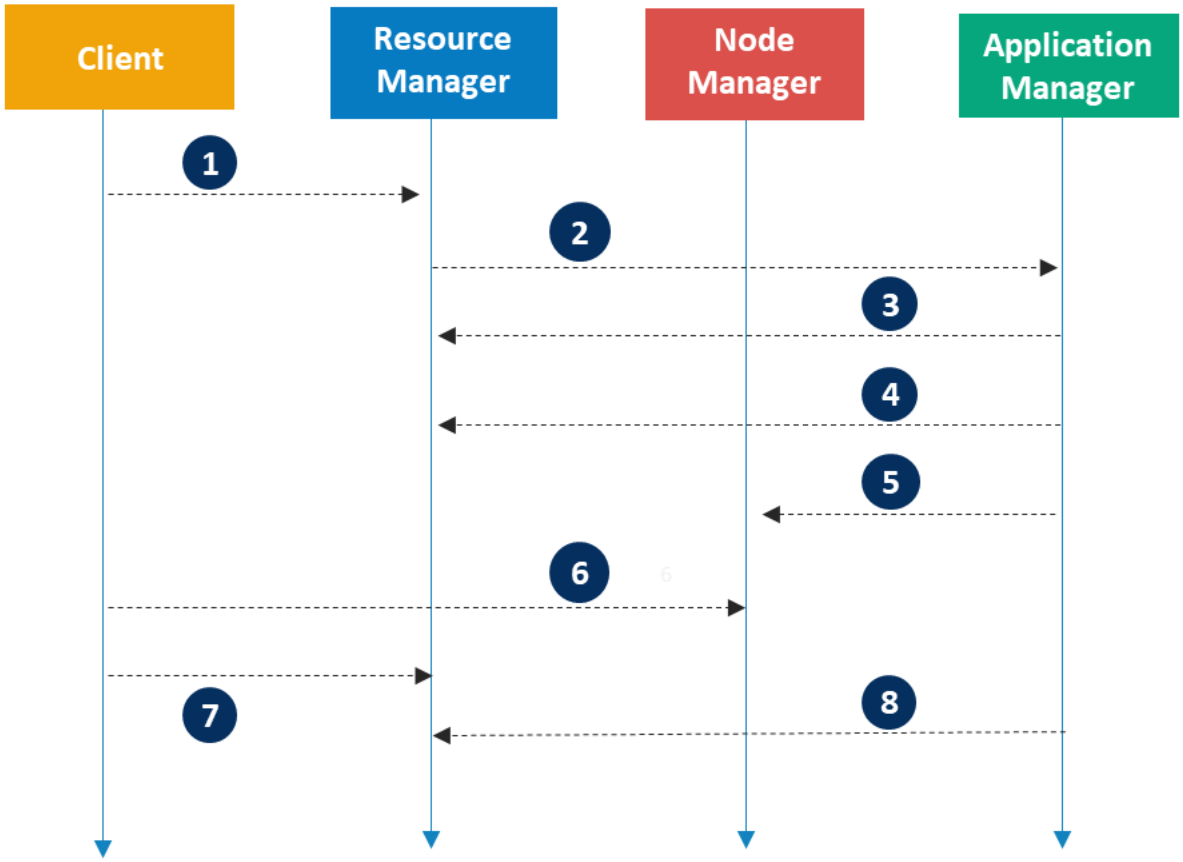
UNIVERSITAT DE BARCELONA

YARN Architecture



UNIVERSITAT DE BARCELONA

YARN Job Workflow



UNIVERSITAT DE BARCELONA

YARN Architecture

1. Client submits an application
2. Resource Manager allocates a container to start Application Manager
3. Application Manager registers with Resource Manager
4. Application Manager asks containers from Resource Manager
5. Application Manager notifies Node Manager to launch containers
6. Application code is executed in the container
7. Client contacts Resource Manager/Application Manager to monitor application's status
8. Application Manager unregisters with Resource Manager

UNIVERSITAT DE BARCELONA

Read at home

- HDFS (Important)
- Watch https://www.youtube.com/watch?v=ziqx2hJY8Hg

- MapReduce
- Watch https://www.youtube.com/watch?v=vbi95iqsnnM