UNIVERSITAT DE BARCELONA

Master in Fundamental Principles of Data Science

Dr Rohit Kumar

Hadoop

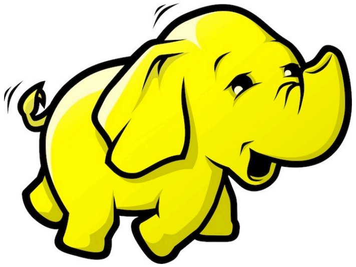
What is Hadoop

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.

Hadoop



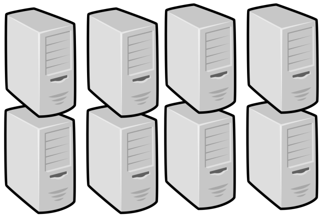
Distributed Computation

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

Distributed Storage

HDFS a distributed file system provides storage

Server cluster

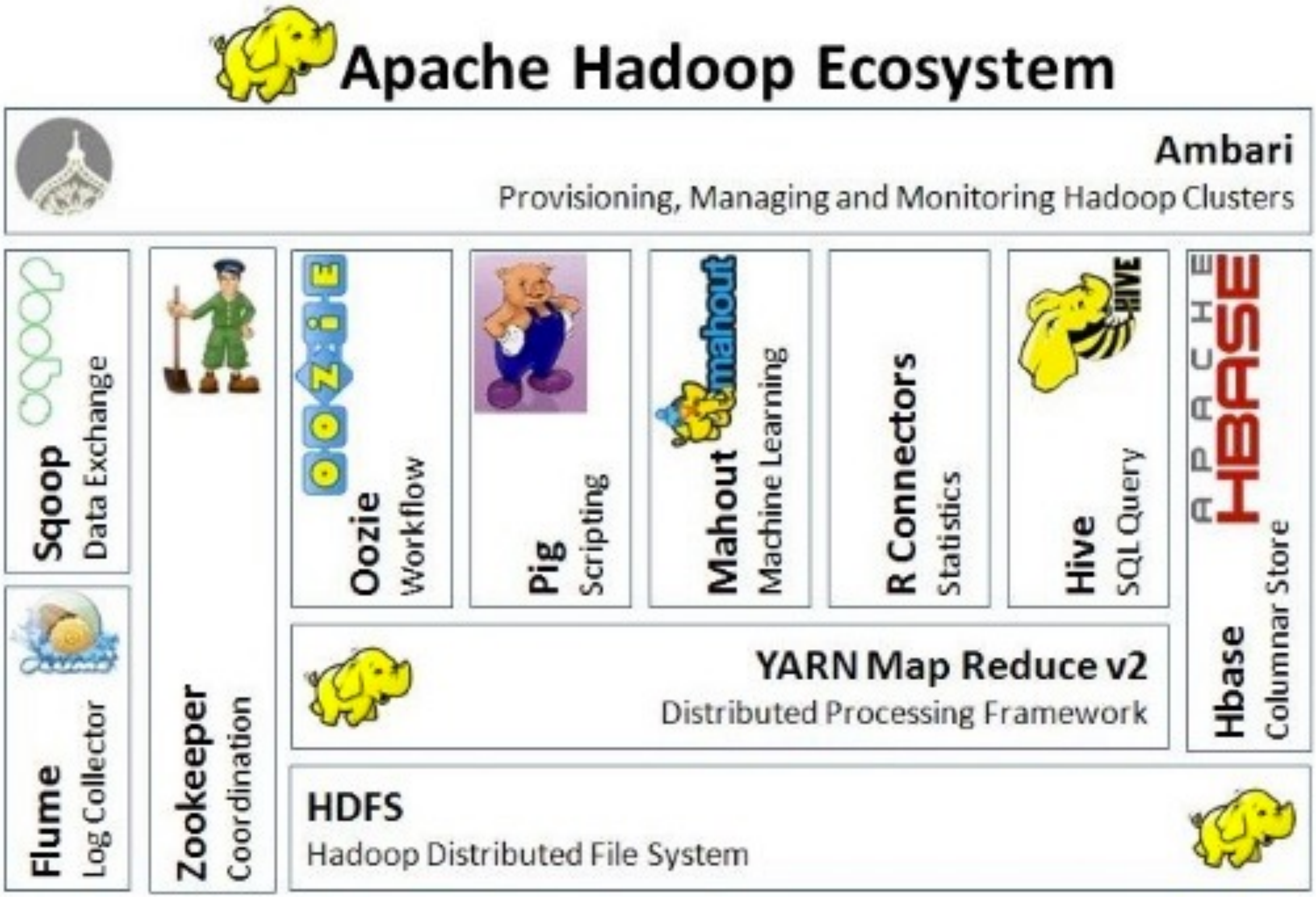


Hadoop can run on commodity hardware

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!

Hadoop EcoSystem



HDFS

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.

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.

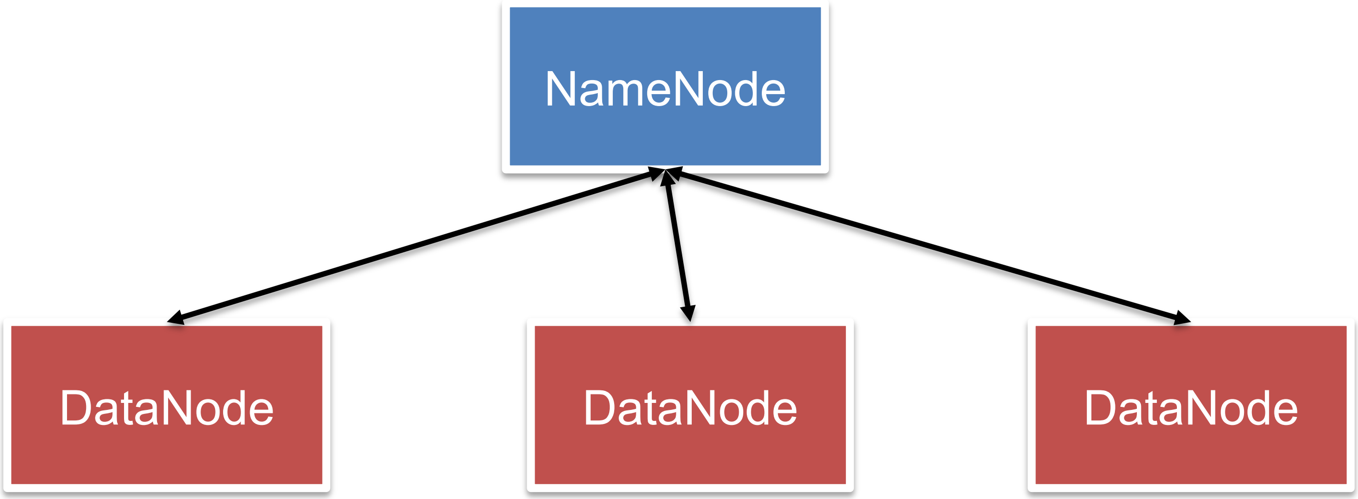
One of the main advantage is **Data Locality**

Data Locality

Data locality talks about moving processing unit to data rather than the data to processing unit. In our traditional system, we used to bring the data to the application layer and then process it. But now, because of the architecture and huge volume of the data, bringing the data to the application layer will reduce the network performance to a noticeable extent. So, in HDFS, we bring the computation part to the data nodes where the data is residing. Hence, you are not moving the data, you are bringing the program or processing part to the data.

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).

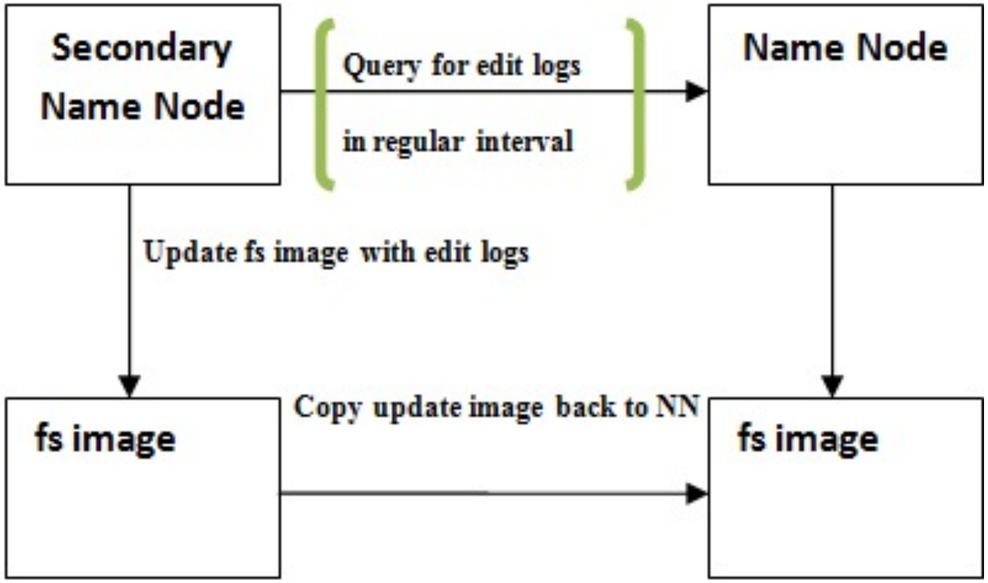


HDFS Architecture

- **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.

Secondary NameNode

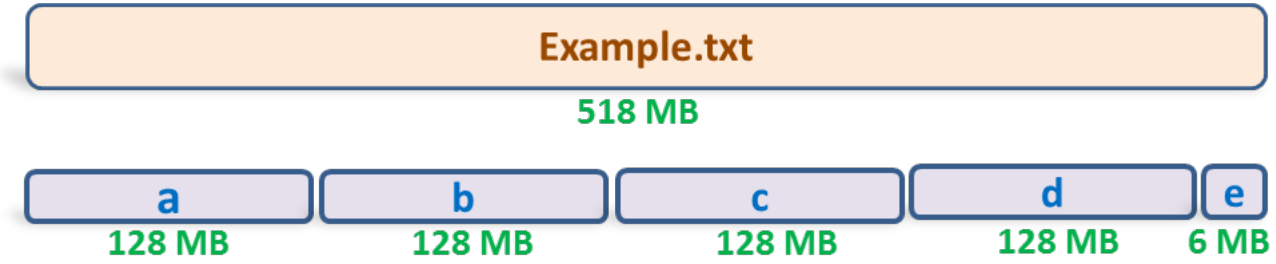
The Secondary NameNode works concurrently with the primary NameNode as a **helper daemon. It is not a backup of NameNode but just a helper.**


Secondary Name Node
Query for edit logs in regular interval
Name Node
Update fs image with edit logs
Copy update image back to NN
fs image
fs image

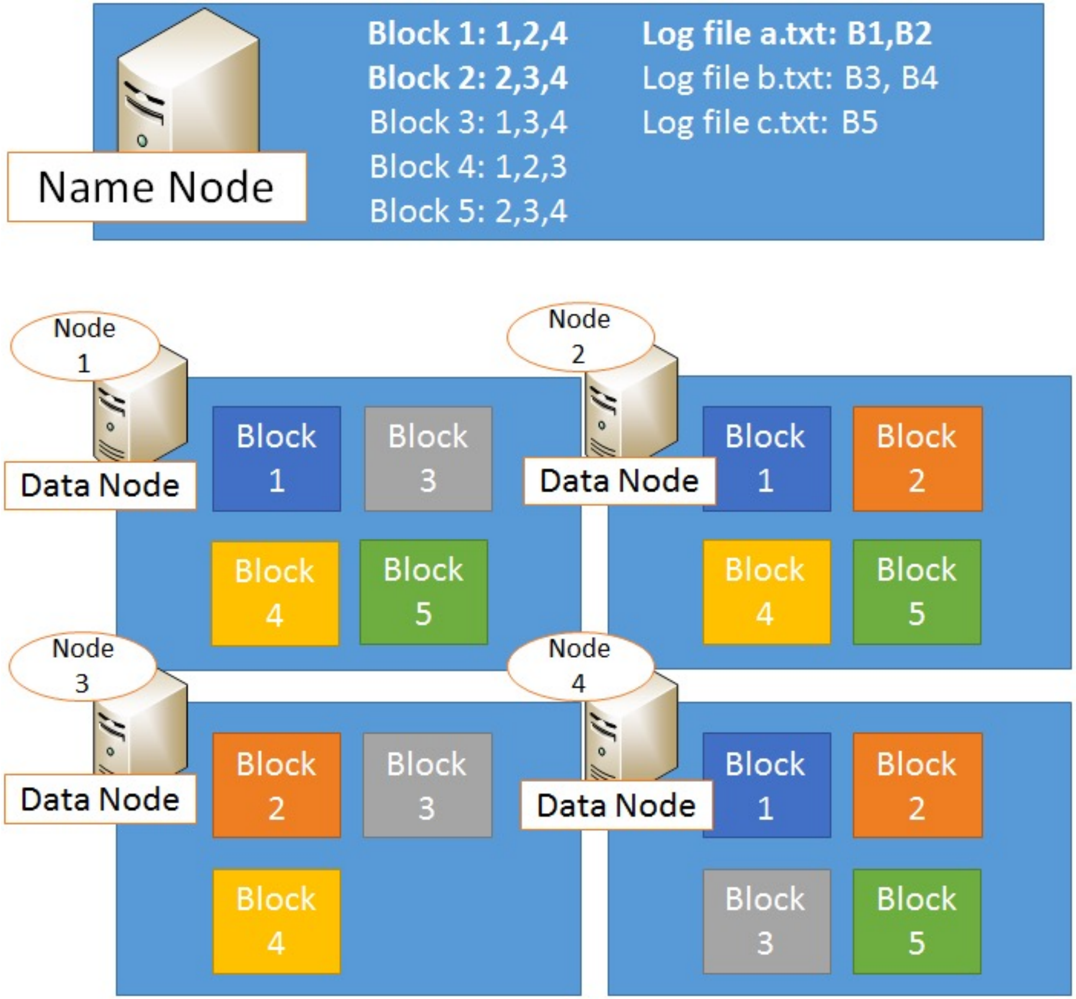
Secondary NameNode performs regular checkpoints in HDFS. Therefore, it is also called CheckpointNode.

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.



Replication management



Replication Factor 3

Hadoop Processing

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*)

One simple approach

Use a Hash Table

“Who is Ram

Ram is who

Ram is king

King of Ajodhya”

Key	Value

One simple approach

Use a Hash Table

“Who is Ram

Ram is who

Ram is king

King of Ajodhya”

Key	Value
Who	1

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

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

One simple approach

Use a Hash Table

“Who is Ram

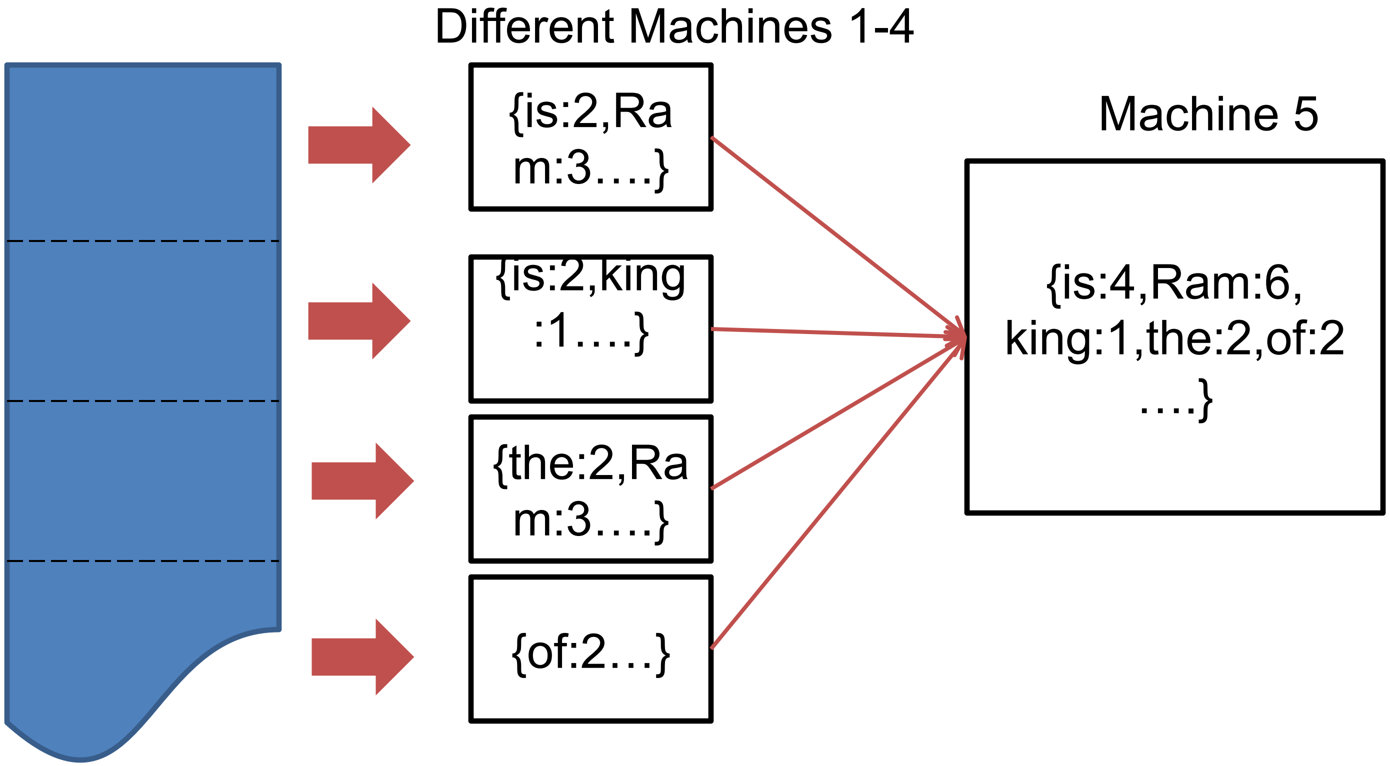
Ram is who

Ram is king

King of Ajodhya”

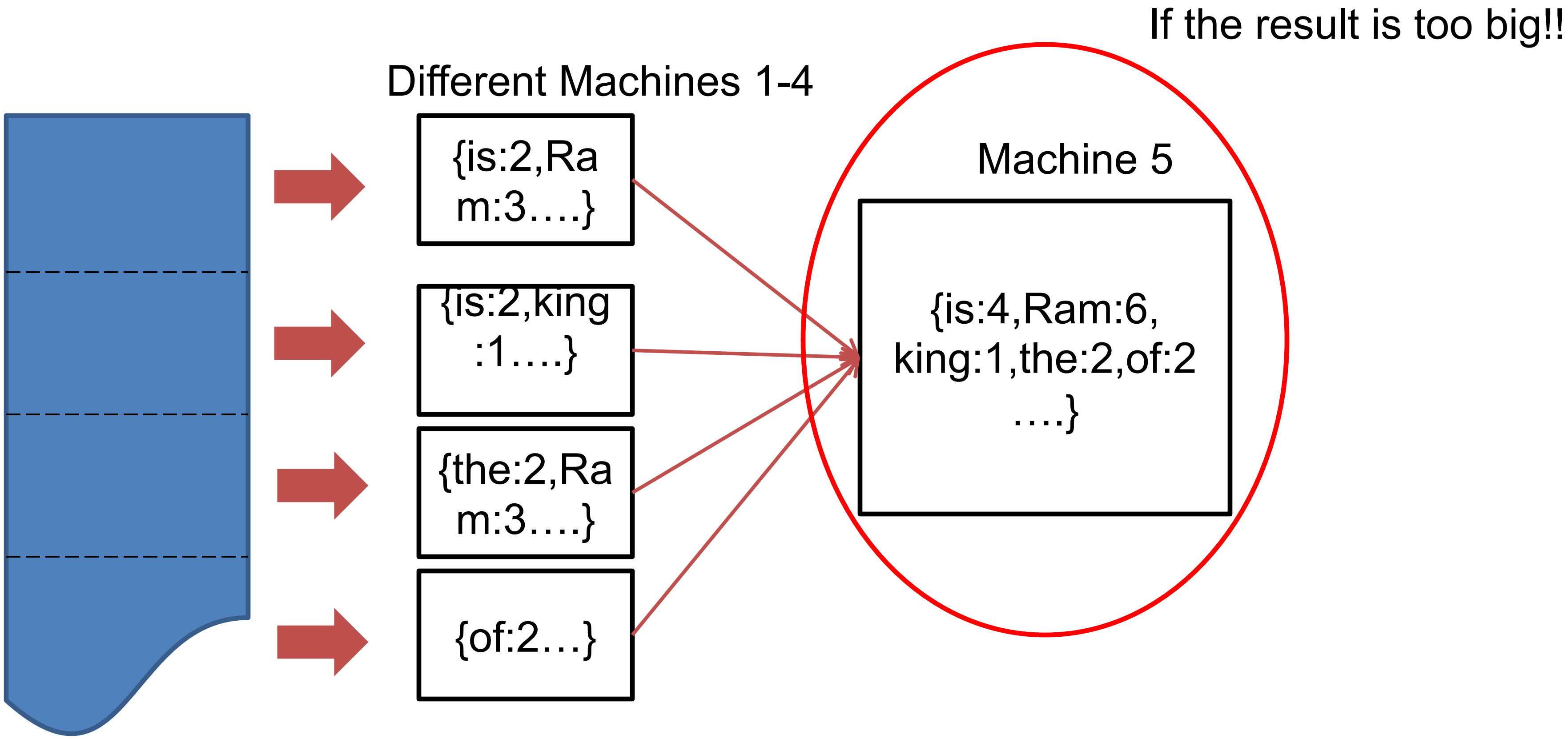
Key	Value
Who	1
Is	1
Ram	2

What if the document is very big?



UNIVERSITAT DE BARCELONA

What if the document is very big?



If the result is too big!!

Different Machines 1-4

{is:2,Ram:3….}

{is:2,king:1….}

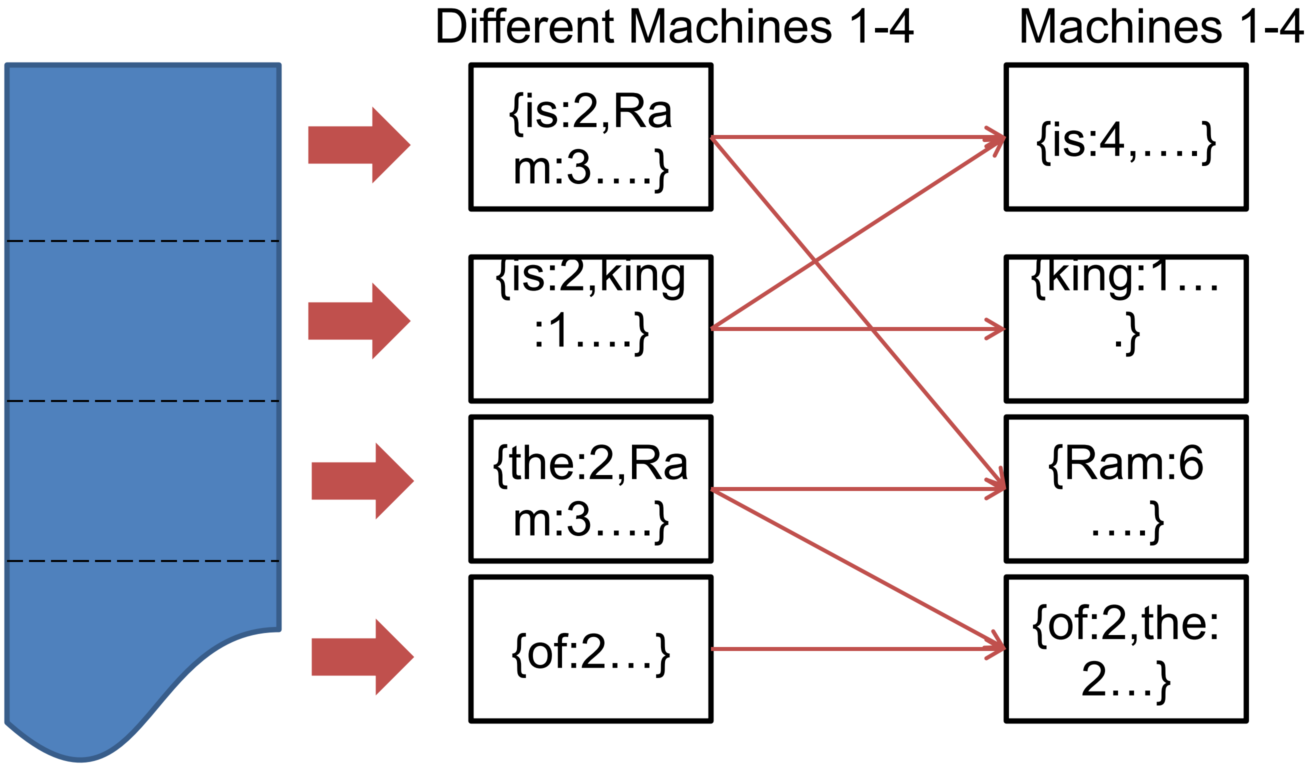
{the:2,Ram:3….}

{of:2…}

Machine 5

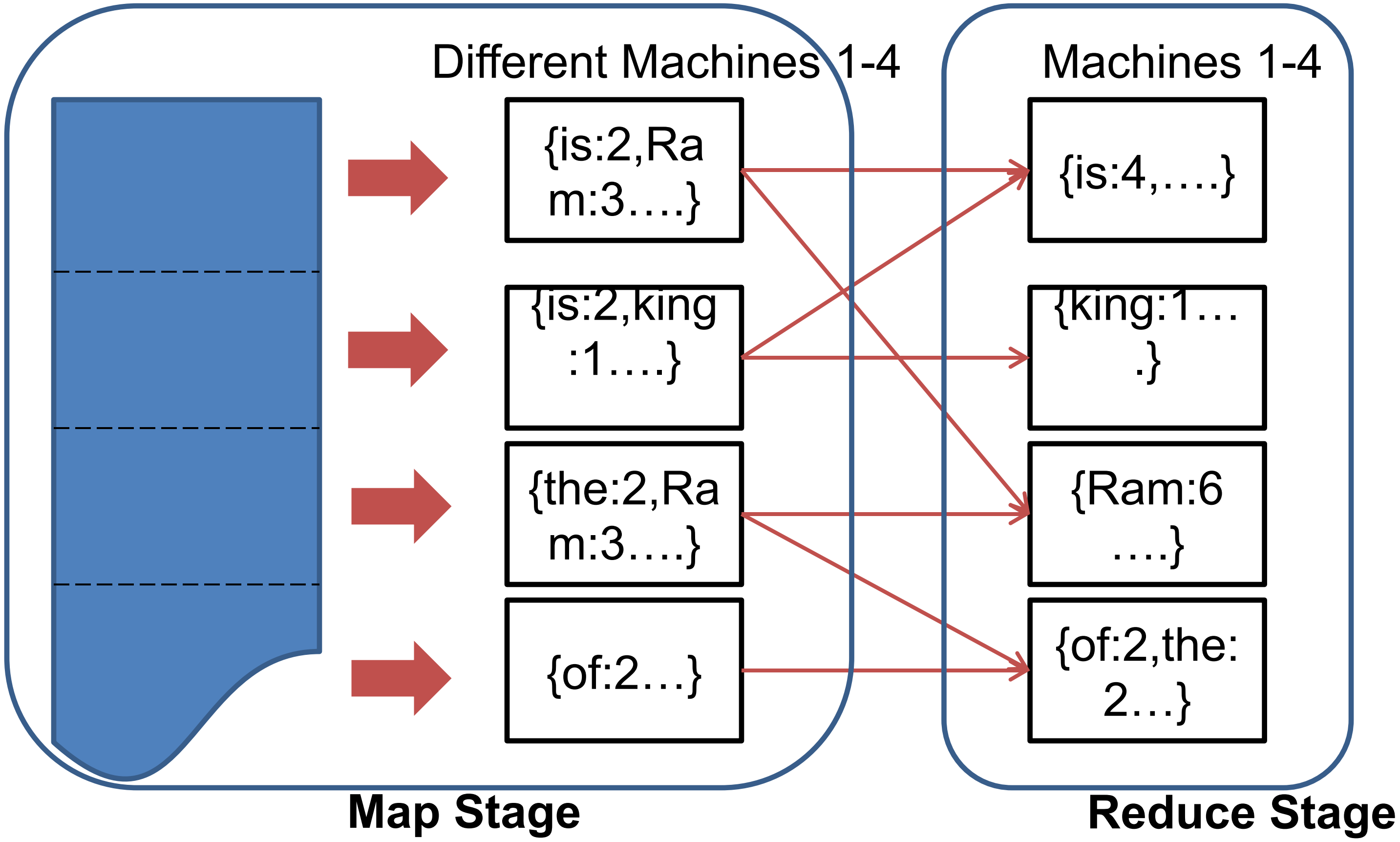
{is:4,Ram:6, king:1,the:2,of:2 ….}

Map Reduce

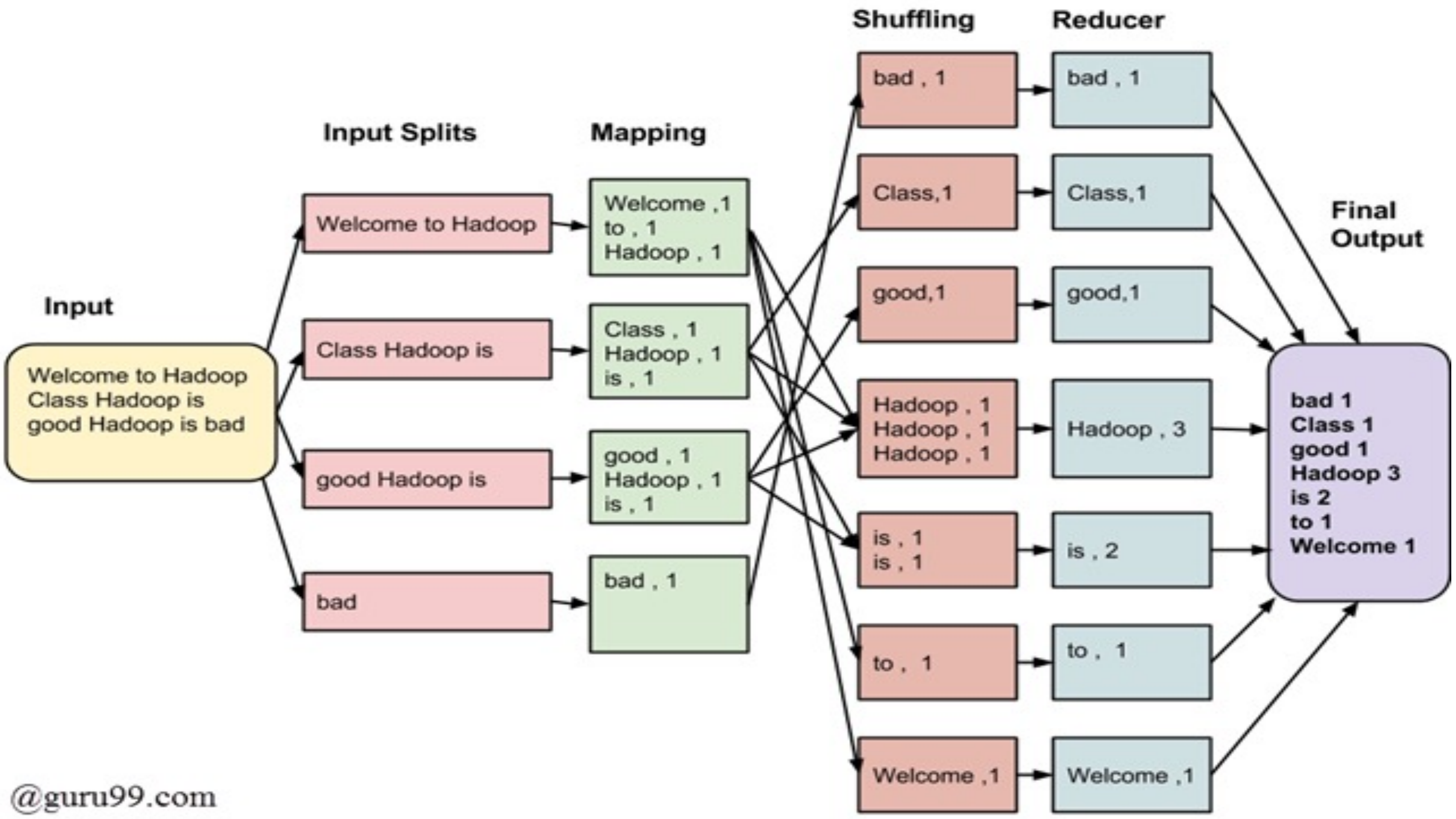

Different Machines 1-4
Machines 1-4
{is:2,Ram:3….}
{is:2,king:1….}
{the:2,Ram:3….}
{of:2…}
{is:4,….}
{king:1….}
{Ram:6….}
{of:2,the:2…}

UNIVERSITAT DE BARCELONA

Map Reduce



Different Phases in MapReduce



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

MapReduce Phases

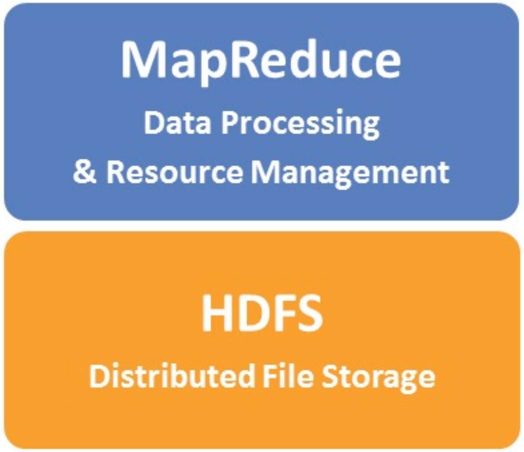
- **Input Splits:** An input to a MapReduce job is divided into fixed-size pieces called **input splits** Input split is a chunk of the input that is consumed by a single map
- **Mapping:** This is the very first phase in the execution of map-reduce program. In this phase data in each split is passed 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>

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.

Hadoop 2.0

Hadoop v1.0



MapReduce

Data Processing

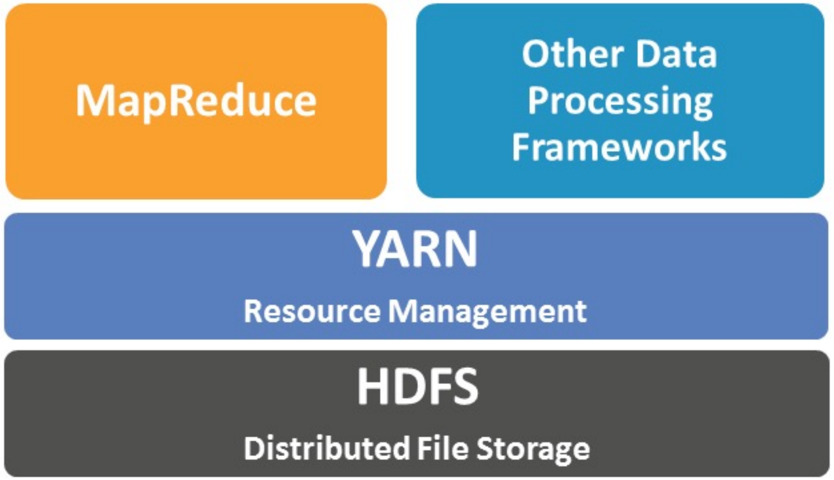
& Resource Management

HDFS

Distributed File Storage

2006-2012

Hadoop v2.0



MapReduce

Other Data Processing Frameworks

YARN

Resource Management

HDFS

Distributed File Storage

2013

Read at home

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

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