Hadoop

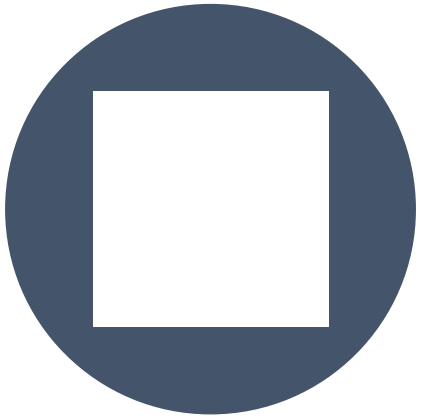
Agenda

INTRODUCTION TO HADOOP

HADOOP DISTRIBUTED FILE SYSTEM

MAPREDUCE FRAMEWORK

Agenda



YARN

OPPORTUNISTIC CONTAINERS

YARN TIMELINE SERVICE V.2

Introduction to Hadoop

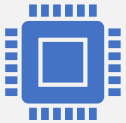
What Is Hadoop ?

Open source software platform for scalable, distributed computing

Provides fast and reliable analysis of both structured data and unstructured data

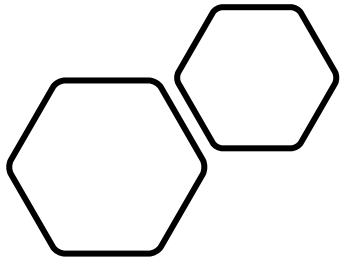
What Is Hadoop ?

Allows for the distributed processing of large datasets across clusters of computers using a simple programming model



Scale up from single servers to thousands of machines, each offering local computation and storage.

Who Uses Hadoop?



A wide variety of companies and organizations use Hadoop for both research and production.

Hadoop Releases

- 1.0.X - current stable version, 1.0 release
- 1.1.X - current beta version, 1.1 release
- 2.X.X - current alpha version
- 0.23.X - simmilar to 2.X.X but missing NN HA.
- 0.22.X - does not include security
- 0.20.203.X - old legacy stable version
- 0.20.X - old legacy version

Modules of Hadoop

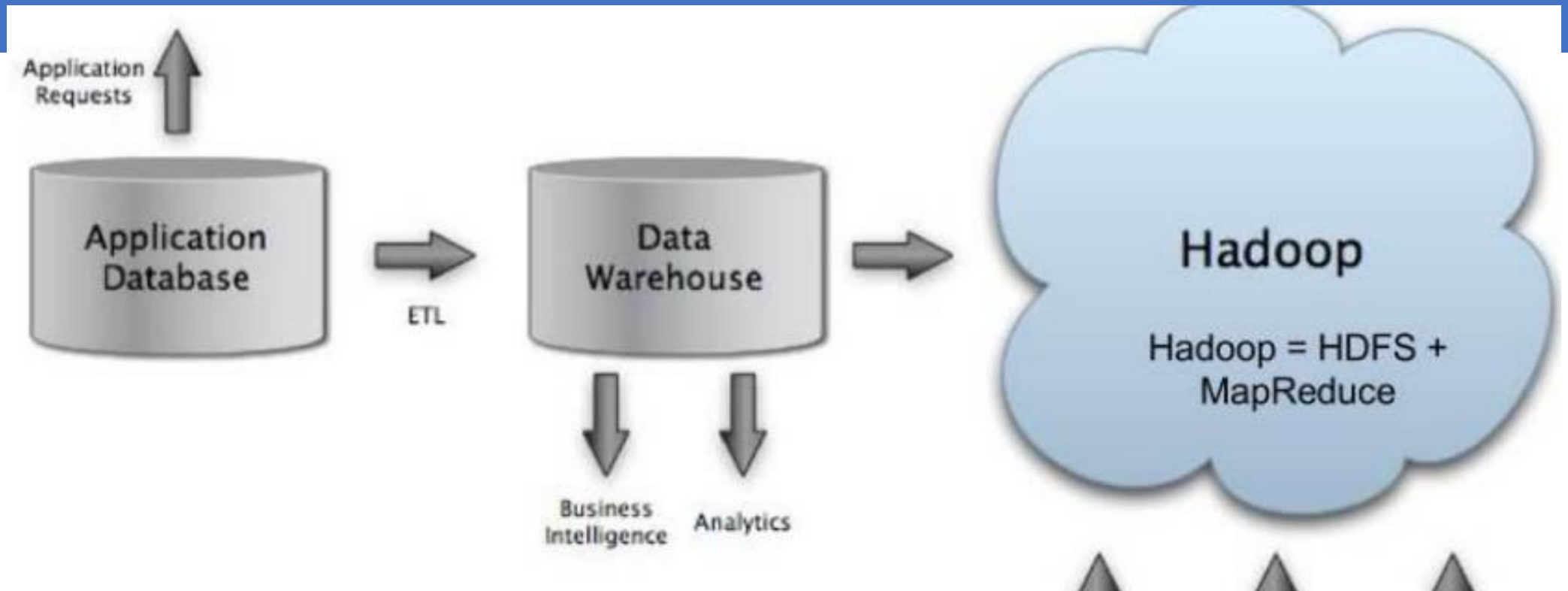
Hadoop Distributed File System(HDFS)

Yet Another Resource Framework (YARN)

Map Reduce

Hadoop Common

Hadoop Architecture



Apache Hadoop = HDFS + MapReduce

- Similar to kernel of an operating system (Hadoop Core)
- Related components are often deployed with Hadoop
- For example: HBase, Hive, Pig, Oozie, Flume, Sqoop
- Together, these components form a "Hadoop Stack"
- Not all components must be deployed

Hadoop Distributed File System(HDFS)

Utilized for storage permission is a Hadoop cluster

Designed for working on commodity Hardware devices

Working on a distributed file system design

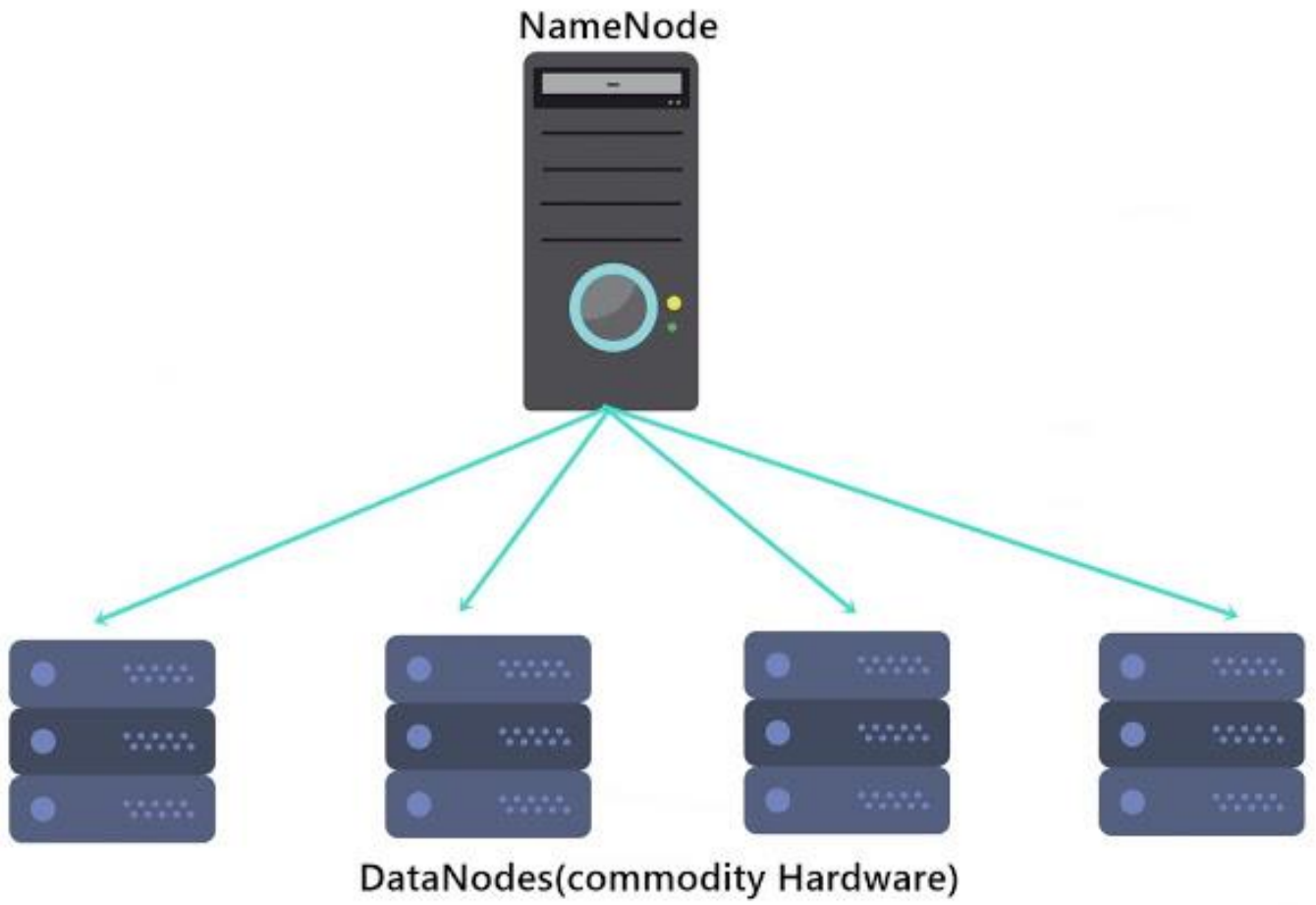
Hadoop Distributed File System(HDFS)

It believes to storing the data in a large chunk of blocks.

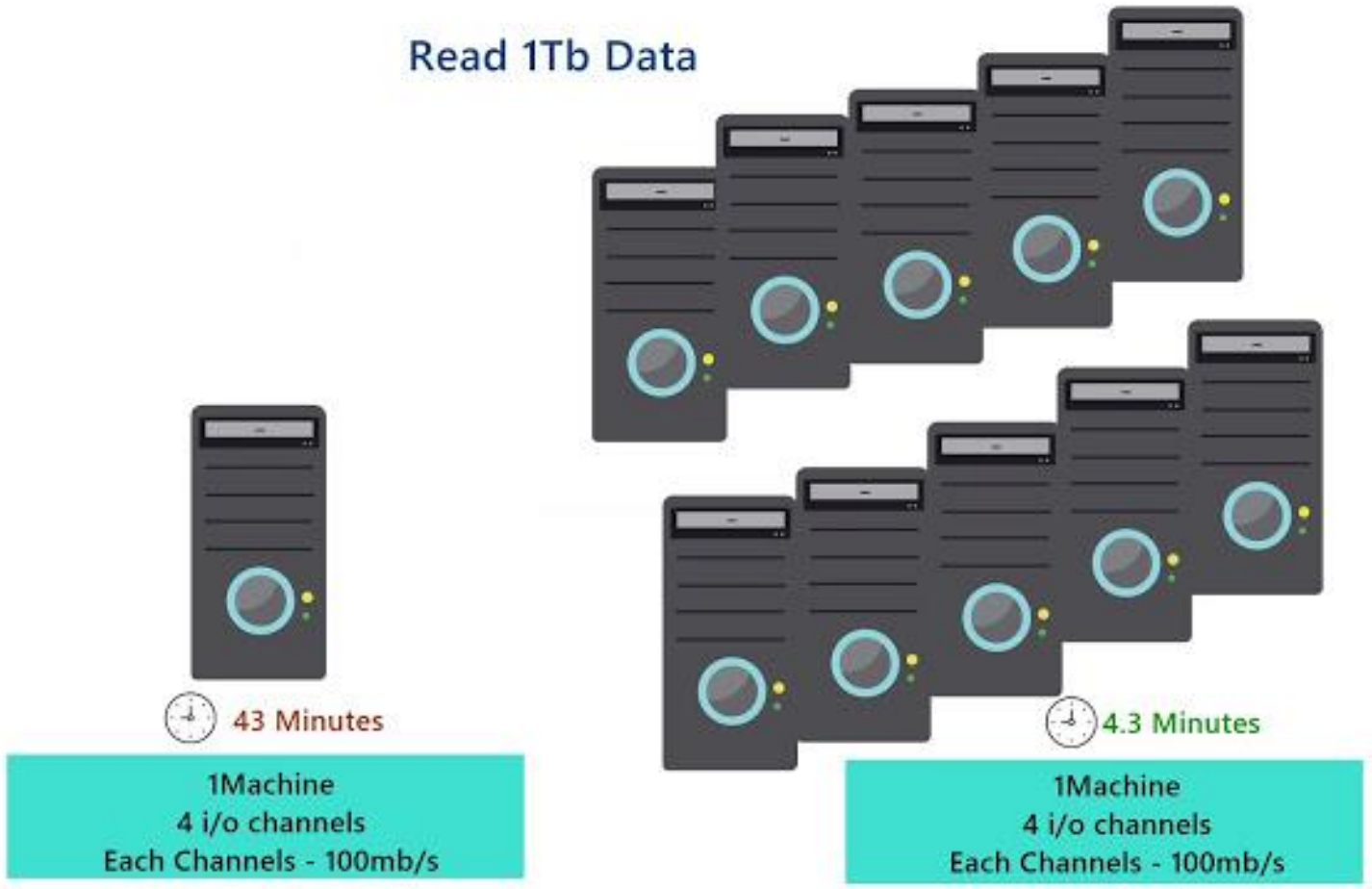
Provides Fault-tolerance and High availability to the storage layer.

Advantages of HDFS

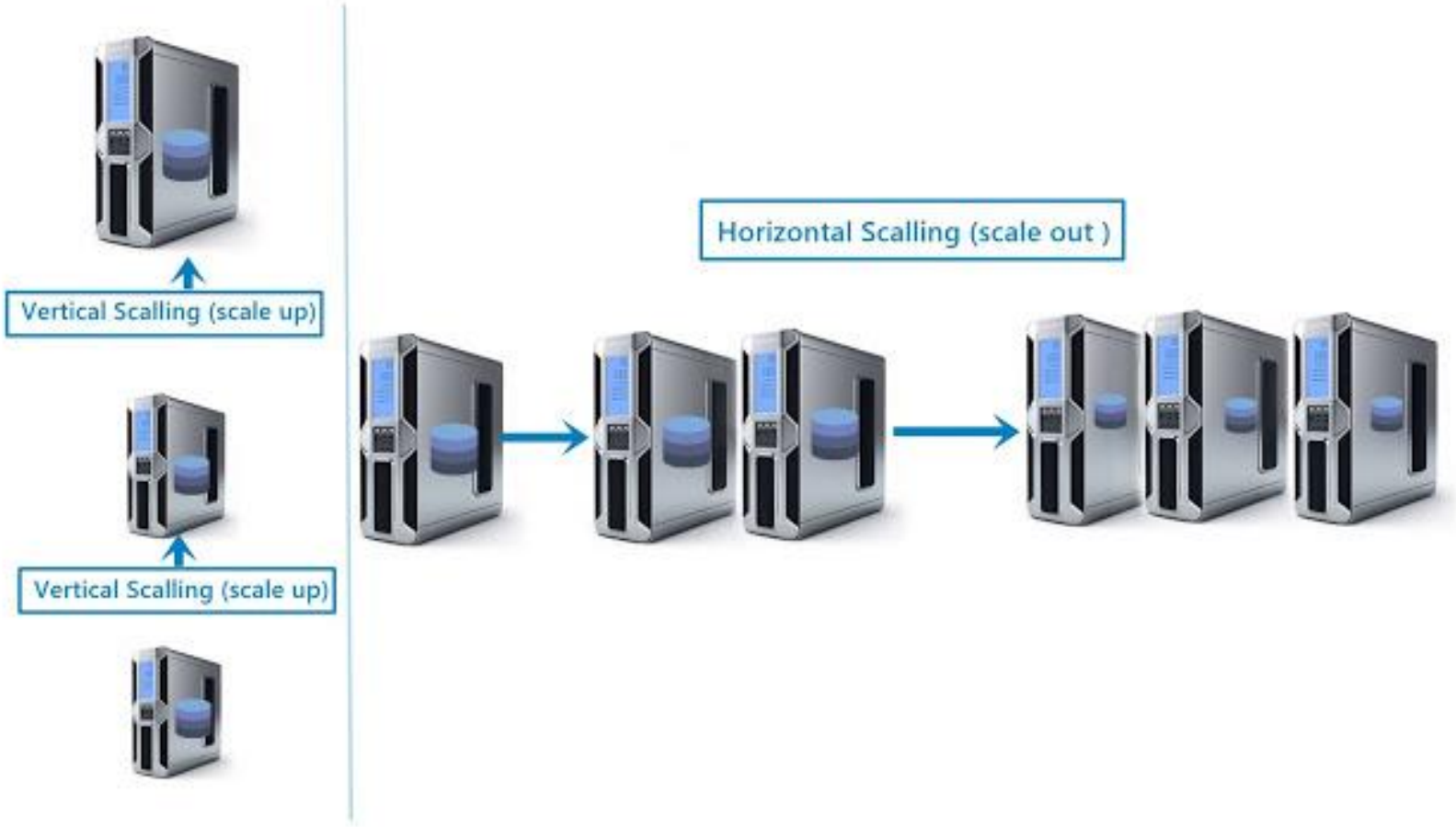
1. Distributed Storage



2 .Horizontal Scalability



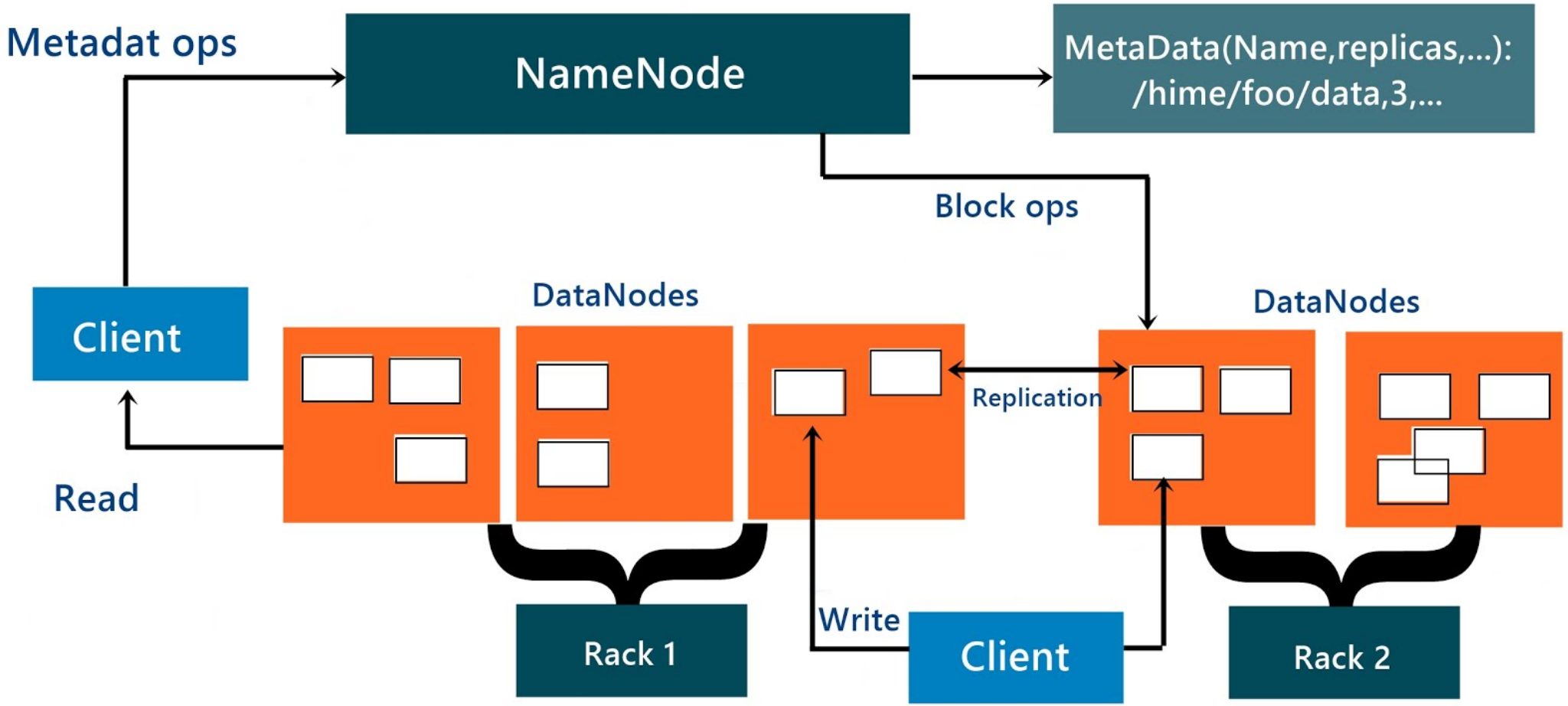
3.Distributed & Parallel Computation



Features of HDFS

- Economical Cost
- **Variety and Volume of Data**
- **Reliability and Fault Tolerance**
- **Data Integrity**
- **High Throughput**
- **Data Locality**

HDFS Architecture



Data storage Nodes in HDFS

NameNode(Master)

DataNode(Slave)

NameNode

Works as a Master in a Hadoop cluster that guides the Datanode(Slaves)

Used for storing the Metadata.

DataNode

Works as a Slave DataNodes are mainly utilized for storing the data in a Hadoop cluster

Number of DataNodes can be from 1 to 500

High storing capacity to store a large number of file blocks.

MapReduce framework

A software framework for distributed processing of large data sets

The framework takes care of scheduling tasks, monitoring them and re-executing any failed tasks.

Splits the input data set into independent chunks that are processed in a completely parallel manner

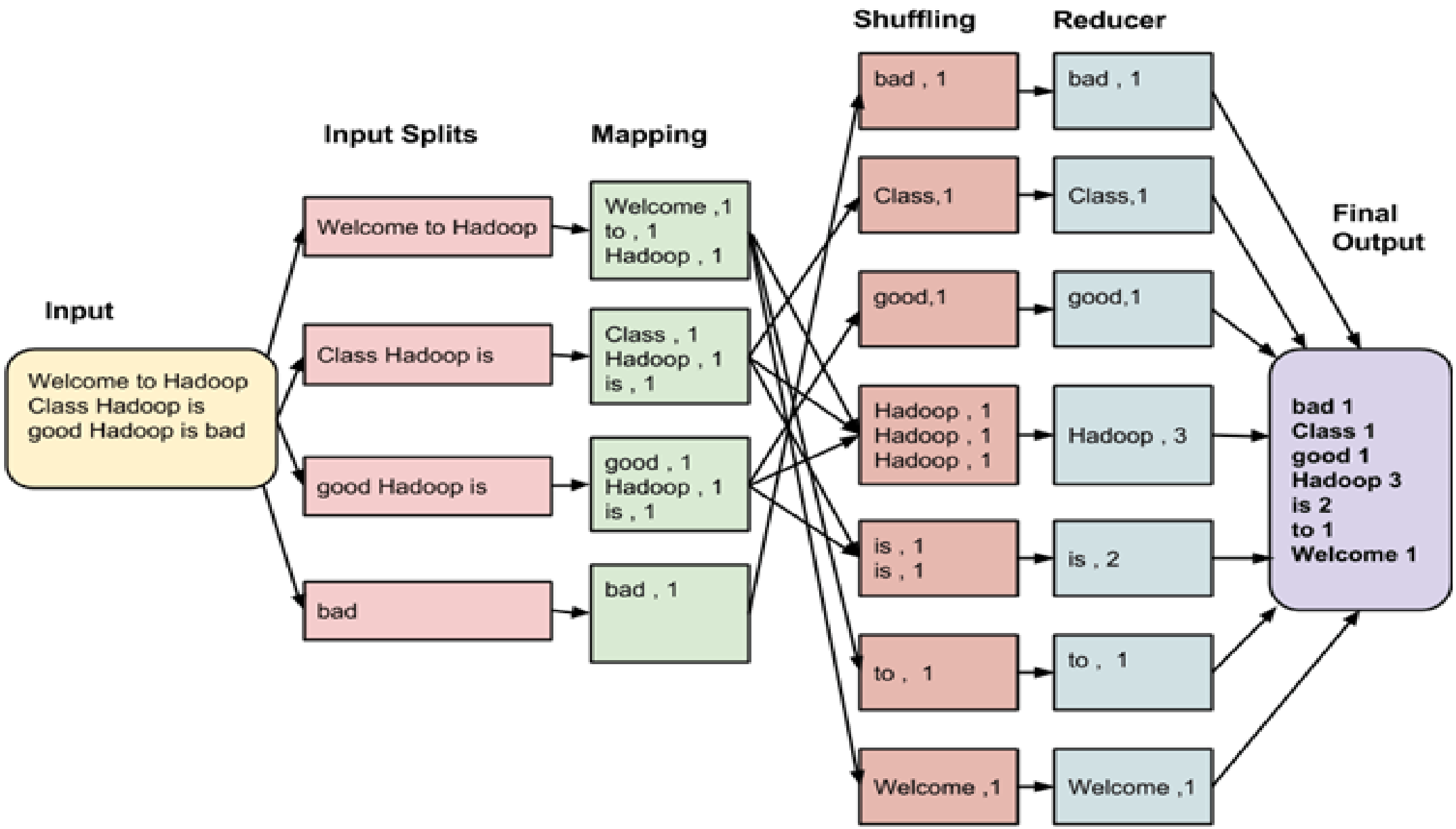
Sorts the outputs of the maps, which are then input to the reduce tasks

Both the input and the output of the job are stored in a file system.

MapReduce example

Consider you have following input data for your MapReduce in Big data Program

```
Welcome to Hadoop Class
Hadoop is good
Hadoop is bad
```


Input
Welcome to Hadoop
Class Hadoop is
good Hadoop is bad
Input Splits
Welcome to Hadoop
Class Hadoop is
good Hadoop is
bad
Mapping
Welcome ,1
to , 1
Hadoop , 1
Class , 1
Hadoop , 1
is , 1
good , 1
Hadoop , 1
is , 1
bad , 1
Shuffling
bad , 1
Class,1
good,1
Hadoop , 1
Hadoop , 1
Hadoop , 1
is , 1
is , 1
to , 1
Welcome ,1
Reducer
bad , 1
Class,1
good,1
Hadoop , 3
is , 2
to , 1
Welcome ,1
Final
Output
bad 1
Class 1
good 1
Hadoop 3
is 2
to 1
Welcome 1

The final output of the MapReduce task is

bad	**1**
Class	1
good	1
Hadoop	3
is	2
to	1
Welcome	1

Phases of MapReduce in Big Data

Input Splits

Mapping

Shuffling

Reducing

YARN

Main component of Hadoop v2.0

Helps to open up Hadoop by allowing to process

YARN

Run data for batch processing, stream processing, interactive processing and graph processing

Helps to run different types of distributed applications other than MapReduce.

Features of YARN

- Multi-tenancy
- Cluster Utilization
- Compatibility
- Scalability

Components of YARN

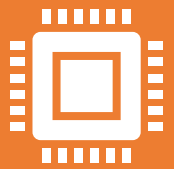
Container

Application Master

Node Manager

Resource Manager

Container



Collection of physical resources such as RAM, CPU cores and disk on a single node.

Invoked by Container Launch Context(CLC)

CLC record that contains information such as environment variables, security tokens, dependencies etc.

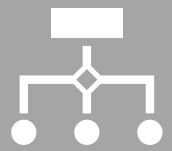
Application Master

Single job submitted to a framework

Application master requests the container from the node manager by sending a Container Launch Context(CLC)

Application Master

CLC which includes everything an application needs to run



Once the application is started, it sends the health report to the resource manager from time-to-time

Node Manager

It take care of individual node on Hadoop cluster

Manages application and workflow and that particular node

It monitors resource usage, performs log management

Node Manager

Kills a container based on directions from the resource manager

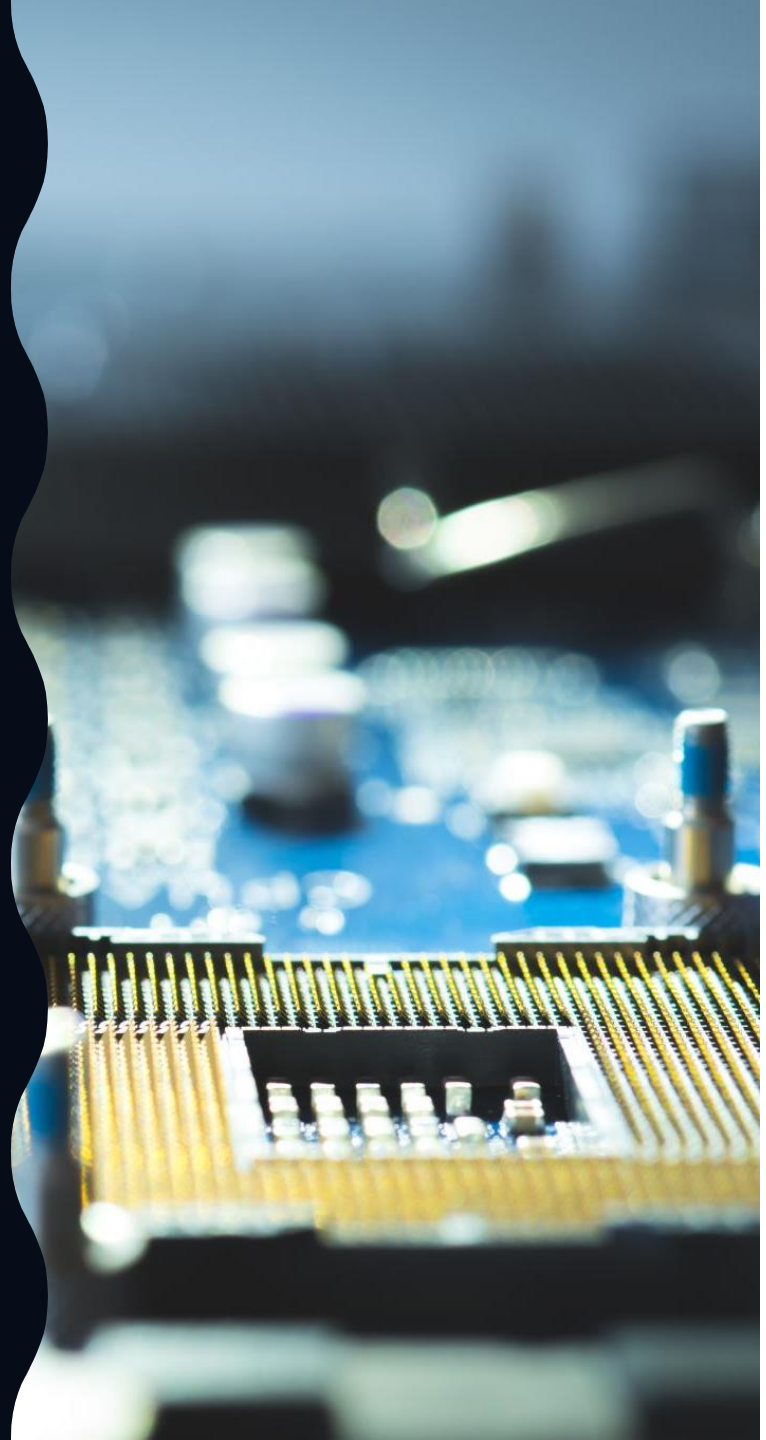
Responsible for creating the container process and start it on the request of Application master.

Resource Manager

Master daemon of YARN

Responsible for resource assignment and management among all the applications.

Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly.



It has two major components

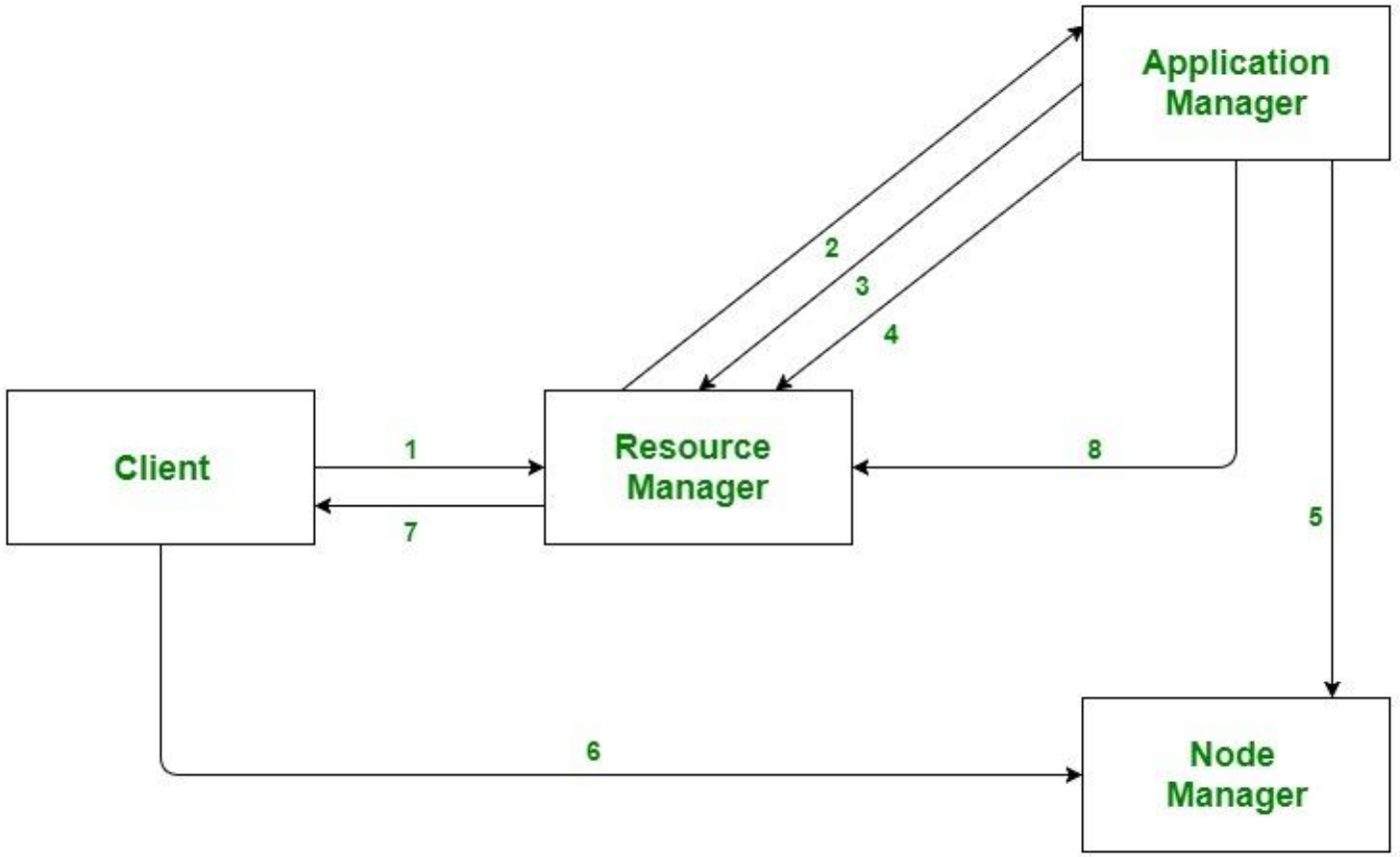
Scheduler

Performs scheduling based on the allocated application and available resources

Application manager

Responsible for accepting the application and negotiating the first container from the resource manager.

Application workflow in YARN:



Application workflow in YARN

Client submits an application

The Resource Manager allocates a container to start the Application Manager

The Application Manager registers itself with the Resource Manager

Application workflow in YARN

5 .The Application Manager negotiates containers from the Resource Manager

6. The Application Manager notifies the Node Manager to launch containers

7. Application code is executed in the container

Application workflow in YARN

8. Client contacts Resource Manager/Application Manager to monitor application's status

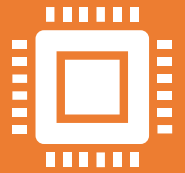
9. Once the processing is complete, the Application Manager un-registers with the Resource Manager

Hadoop Common

Collection of common utilities and libraries that support other Hadoop modules.

Essential part or module of the Apache Hadoop Framework

Hadoop Common



Along with the Hadoop Distributed File System (HDFS), Hadoop YARN and Hadoop MapReduce.

Assumes that hardware failures are common and that these should be automatically handled in software by the Hadoop Framework.

Opportunistic containers

Containers are allocated to nodes by the scheduler only when there is sufficient unallocated resources at a node

Execution of a container will only be completed if there is no violation of fairness or capacity.

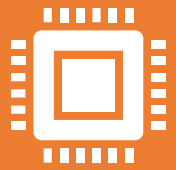
The current container execution design allows an efficient task execution

YARN Timeline Service v.2

YARN Timeline Service v.2 is the next major iteration of Timeline Server

Responsible for collecting different metric data through its timeline collectors

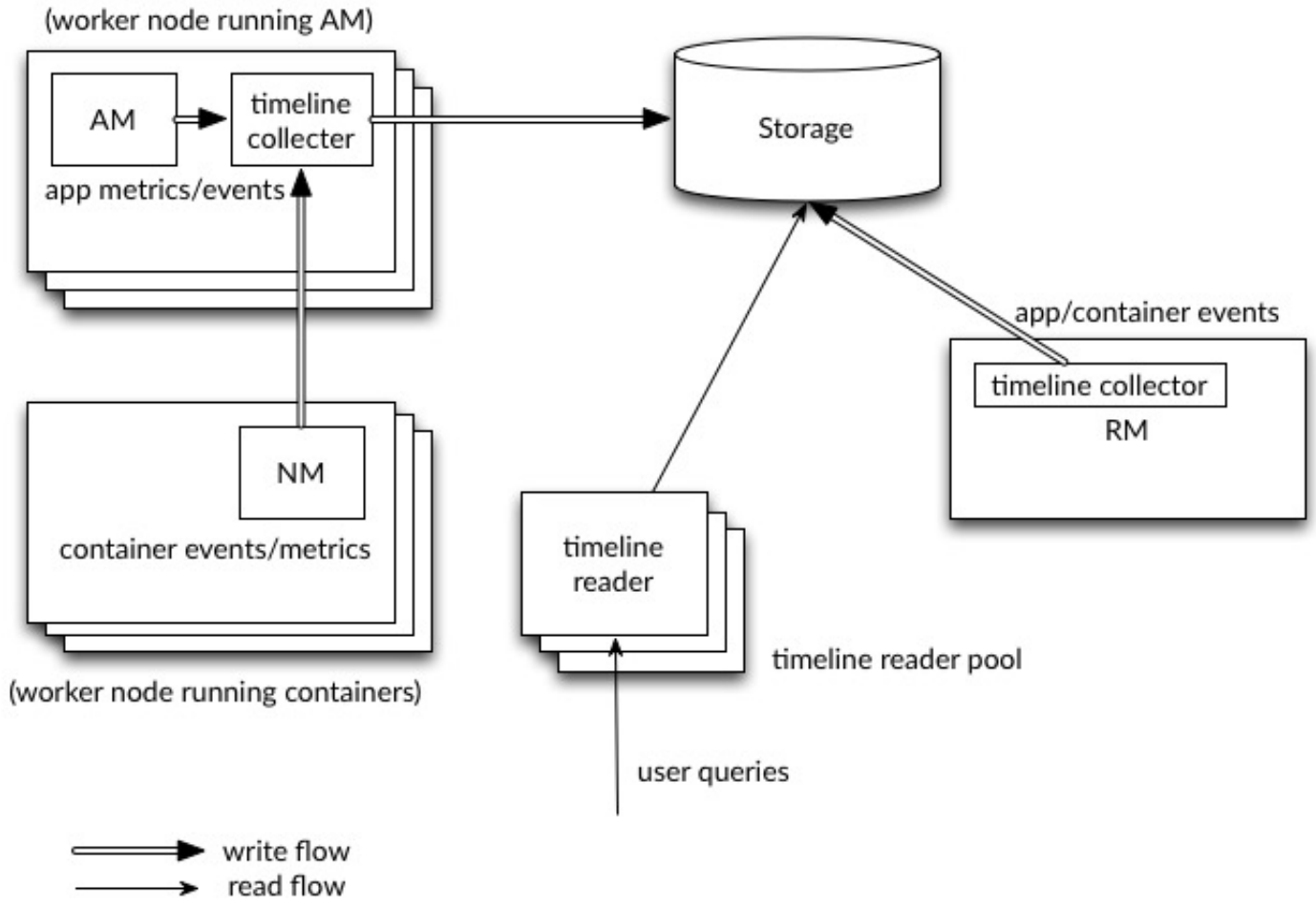
YARN Timeline Service v.2



Provides a RESTful API service to allow users to run queries for getting insights on this data.

It supports aggregation of information.

The following diagram illustrates the design at a high level.



Thank You !