Chapter 2: Introduction to Hadoop & Hadoop Architecture

By Ms. Tina D’abreo

Content

- **Big Data – Apache Hadoop & Hadoop EcoSystem**
- Moving Data in and out of Hadoop
  - Understanding inputs and outputs of MapReduce Concept of Hadoop
- HDFS Commands
- MapReduce
  - The Map Tasks
  - Grouping by Key
  - The Reduce Tasks
  - Combiners
  - Details of MapReduce Execution

Hadoop - Overview

- Hadoop is ***open source*** software platform for ***distributed storage*** and ***distributed processing*** very ***large data sets*** on ***computer clusters*** built from commodity hardware.
- ***Distributed storage*** is that you can just keep on adding more and more computers to your cluster and their hard drives will just become part of your data storage for your data and what you do gives you a way of viewing all of the data distributed across all of the hard drives in your cluster as one single file system.
- The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell scripts

Hadoop - History

- Yahoo was building “Nutch”- an open source search engine - project taken up by Doug Cutting and Mike Cafarella in 2002
- Google published GFS 2003 addressing data storage concern and MapReduce papers in 2004 for data processing improvisation
- In 2005, NDFS - Nutch Distributed File System was invented.
- In 2006, Doug Cutting and Mike Cafarella released new project named Hadoop with the file system HDFS. Hadoop was named after the toy elephant of Doug’s son
- In 2008, Hadoop became the fastest system to sort 1 terabyte of data on a 910 node cluster in 209 seconds.
- GFS is basically what inspired Hadoop's distributed data storage and mapreduce is what inspired Hadoop's distributed processing.

Hadoop - History

- Doug Cutting with the stuffed toy - Hadoop

Why Hadoop? Or Importance of Hadoop

- **Stores and processes** humongous data at a faster rate. The data may be structured, semi-structured, or unstructured
- Protects application and data processing against **hardware failures**. Whenever a node gets down, the processing gets redirected automatically to other nodes and ensures running of applications
- Organizations can store raw data and processor filter it for specific analytic uses as and when required
- As Hadoop is **scalable,** organizations can handle more data by adding more nodes into the systems
- Supports real-time analytics, drives better operational decision-making and **batch workloads** for historical analysis

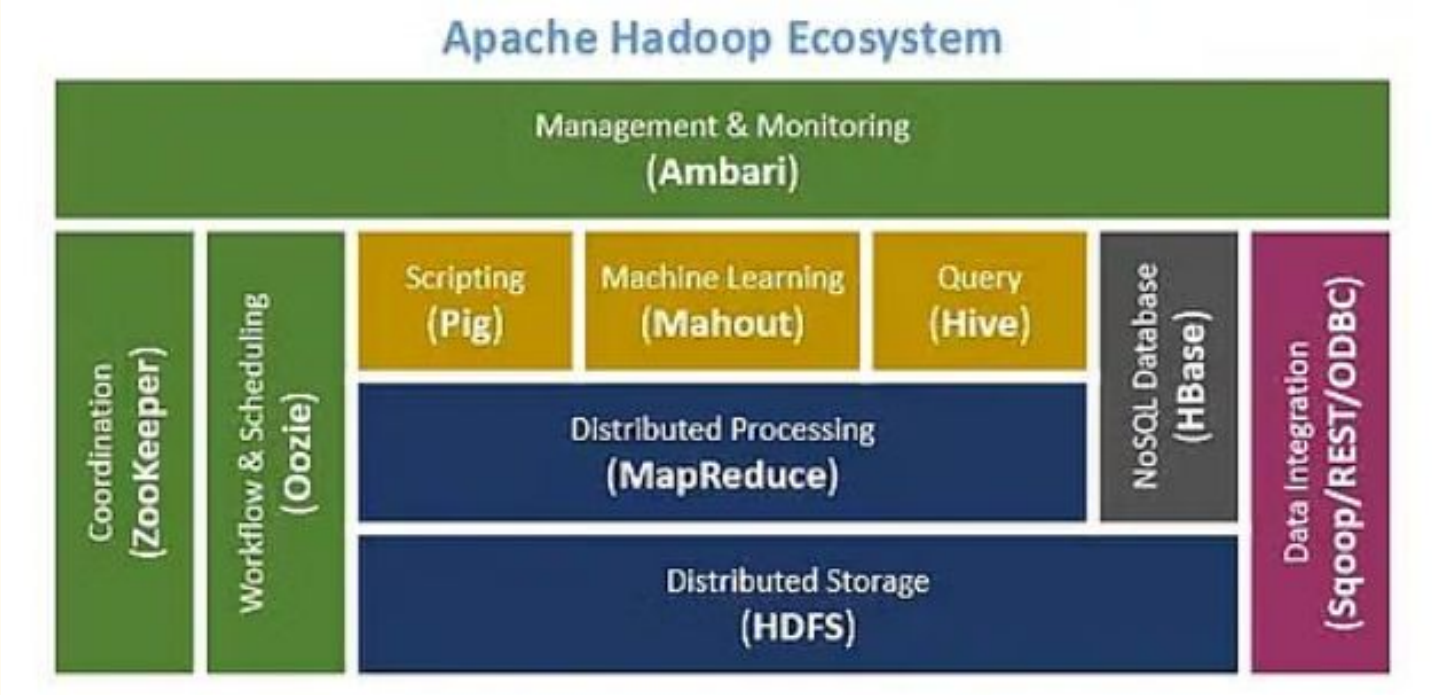
Hadoop - It's Advantages

- Scalable - storage with increasing clusters
- Fault Tolerance / Resilience to Failure
- Economical - commodity software readily available s/ws
- Flexibility - working with unstructured data and developers can use different programming languages.
- Unlimited Data storage and Data computing power
- Cloud services

Hadoop Ecosystem

- Hadoop is a framework which allows us to store and process large data sets in parallel and distributed fashion
- Hadoop Ecosystem encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.

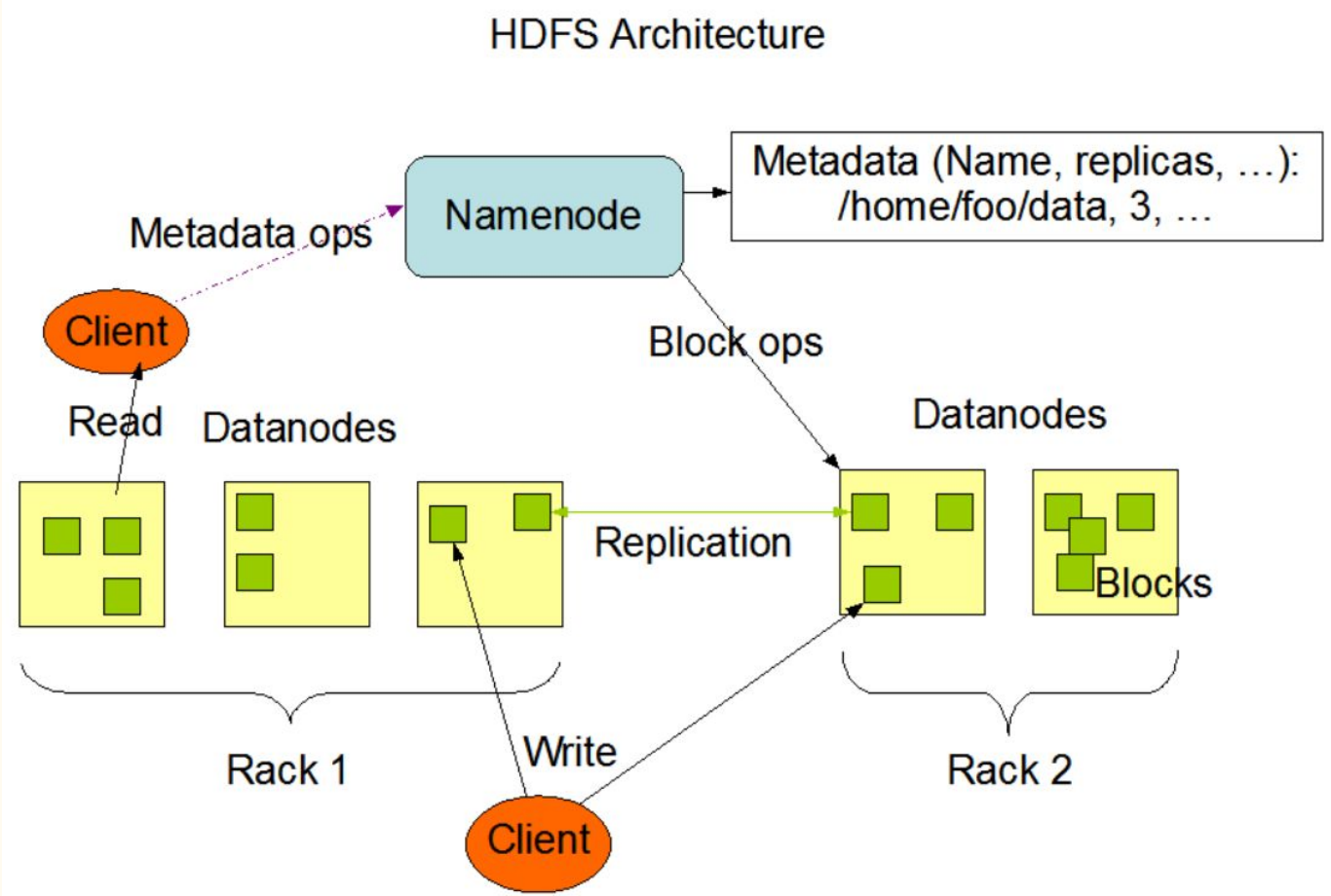
Hadoop Ecosystem



Hadoop Ecosystem - HDFS

- Hadoop Distributed File System
- HDFS is designed to provide greater **scalability and fault resistance** to the Hadoop framework.
- The primary function of HDFS is to act as a **storage backend** for other distributed applications, including MapReduce and HBase.
- HDFS is a component of the Hadoop framework used for **storing and managing large datasets** across several connected computers.

Hadoop Ecosystem - HDFS Architecture



Hadoop Ecosystem - HDFS Architecture

- HDFS is mainly designed for working on commodity Hardware devices(inexpensive devices), working on a distributed file system design.
- HDFS has two important components: NameNode and DataNode
- An HDFS cluster consists of a **single NameNode and multiple DataNodes**
- ***NameNode:***
  - A master server that manages the file system namespace and regulates access to files by clients.
  - NameNode executes file system namespace operations like opening, closing, and renaming files and directories.
  - It contains metadata, just like a log file or you can say as a table of content.
  - Hence, it requires less storage and high computational resources.
  - NameNode maintains the **file system namespace**

Hadoop Ecosystem - HDFS Architecture

- ***DataNode:***
  - Manage storage attached to the nodes that they run on.
  - All data is stored on the DataNodes and hence it requires more storage resources.
- ***Rack:***
  - The rack is nothing but just the physical collection of nodes in our Hadoop cluster (maybe 30 to 40).
  - A large Hadoop cluster is consists of multiple Racks.
  - With the help of this Racks information, Namenode chooses the closest DataNode to achieve the maximum performance while performing the read/write information which reduces the Network Traffic.

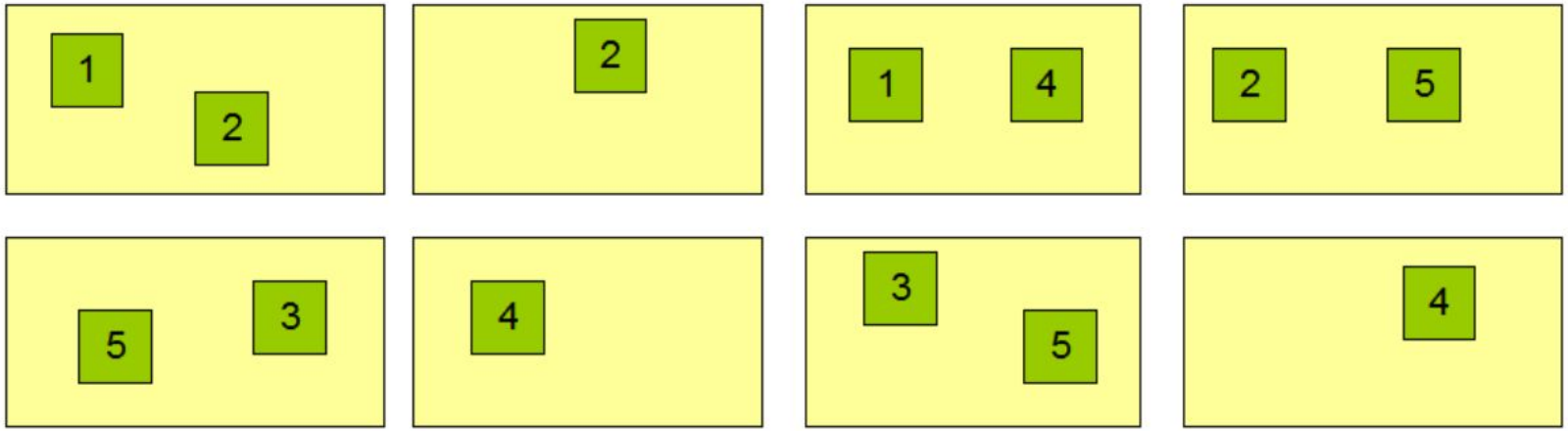
Hadoop Ecosystem - HDFS Architecture

- ***Data Replication:***
  - Ensures availability of data
  - Replication factor is the number of times a copy of something (here, data block) is made.
  - The blocks of a file are replicated for fault tolerance.
  - The block size and replication factor are configurable per file.
- ***File System Namespace:***
  - The file system namespace hierarchy is similar to most other existing file systems;
    - one can create and remove files, move a file from one directory to another, or rename a file.
  - HDFS supports user quotas and access permissions.
  - Any change to the file system namespace or its properties is recorded by the NameNode.
  - An application can specify the number of replicas of a file that should be maintained by HDFS.
  - This information is stored by the NameNode.

Block Replication

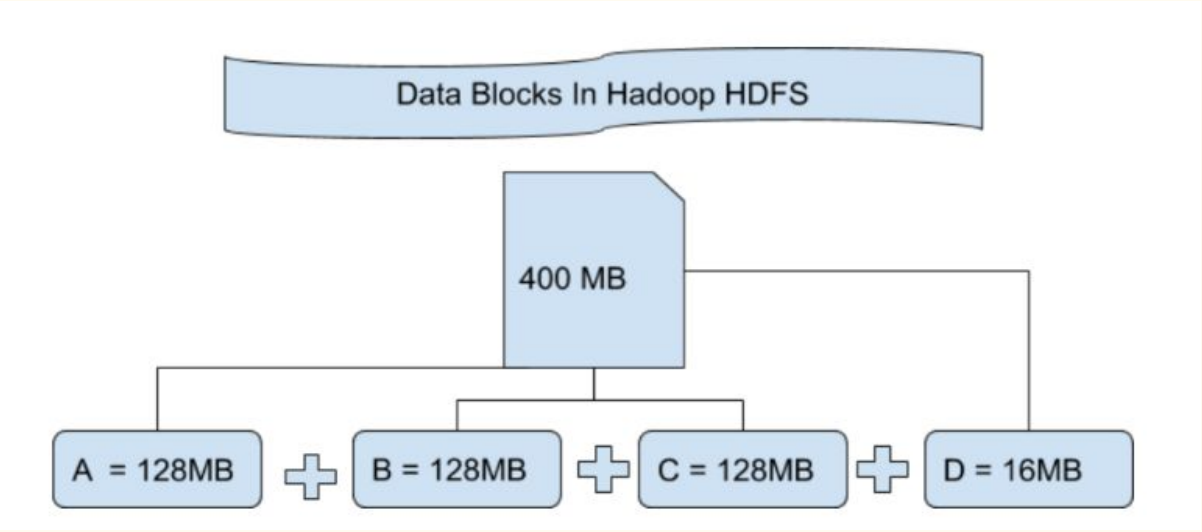
Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes



Hadoop Ecosystem - HDFS Architecture

- ***File Block In HDFS:***
  - Data in HDFS is always stored in terms of blocks.
  - So the single block of data is divided into multiple blocks of size 128MB which is default and you can also change it manually.



Hadoop Ecosystem - HDFS Architecture

- ***Data Replication:***
  - Ensures availability of data
  - Replication factor is the number of times a copy of something (here, data block) is made.
  - The blocks of a file are replicated for fault tolerance.
  - The block size and replication factor are configurable per file.
- ***File System Namespace:***
  - The file system namespace hierarchy is similar to most other existing file systems;
    - one can create and remove files, move a file from one directory to another, or rename a file.
  - HDFS supports user quotas and access permissions.
  - Any change to the file system namespace or its properties is recorded by the NameNode.
  - An application can specify the number of replicas of a file that should be maintained by HDFS.
  - This information is stored by the NameNode.

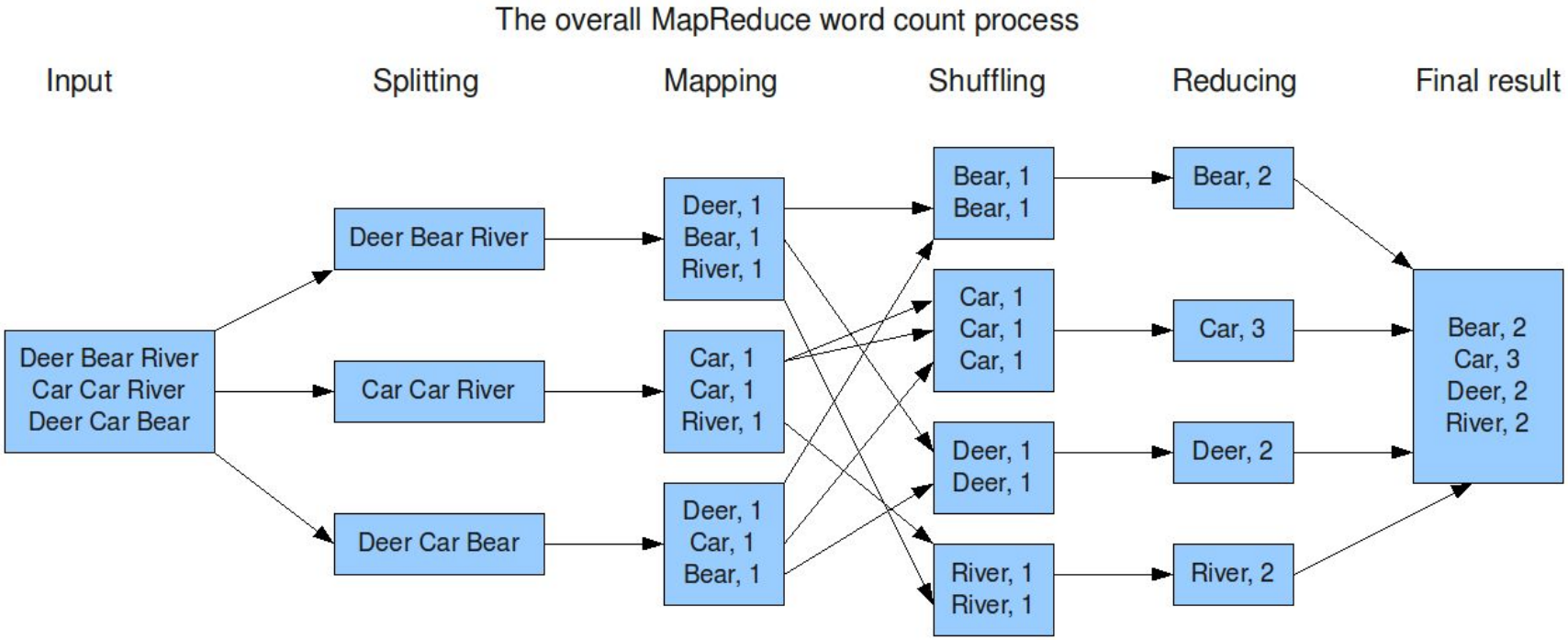
Hadoop Ecosystem - YARN

- YARN is an acronym for **Yet Another Resource Negotiator.**
- YARN is considered as the brain of Hadoop Ecosystem. It performs all your processing activities by **allocating resources and scheduling tasks.**
- It handles the cluster of nodes and acts as Hadoop's **resource management unit.**
- YARN **allocates** RAM, memory, and other resources to different applications.
- 2 major components of YARN:
    - **ResourceManager and NodeManager**
        - ***ResourceManager:*** main node in the processing department.
        - It receives the processing requests, and then passes the parts of requests to corresponding Node Managers accordingly, where the actual processing takes place.
        - ***NodeManagers:***
        - they are installed on every DataNode. It is responsible for execution of task on every single DataNode.

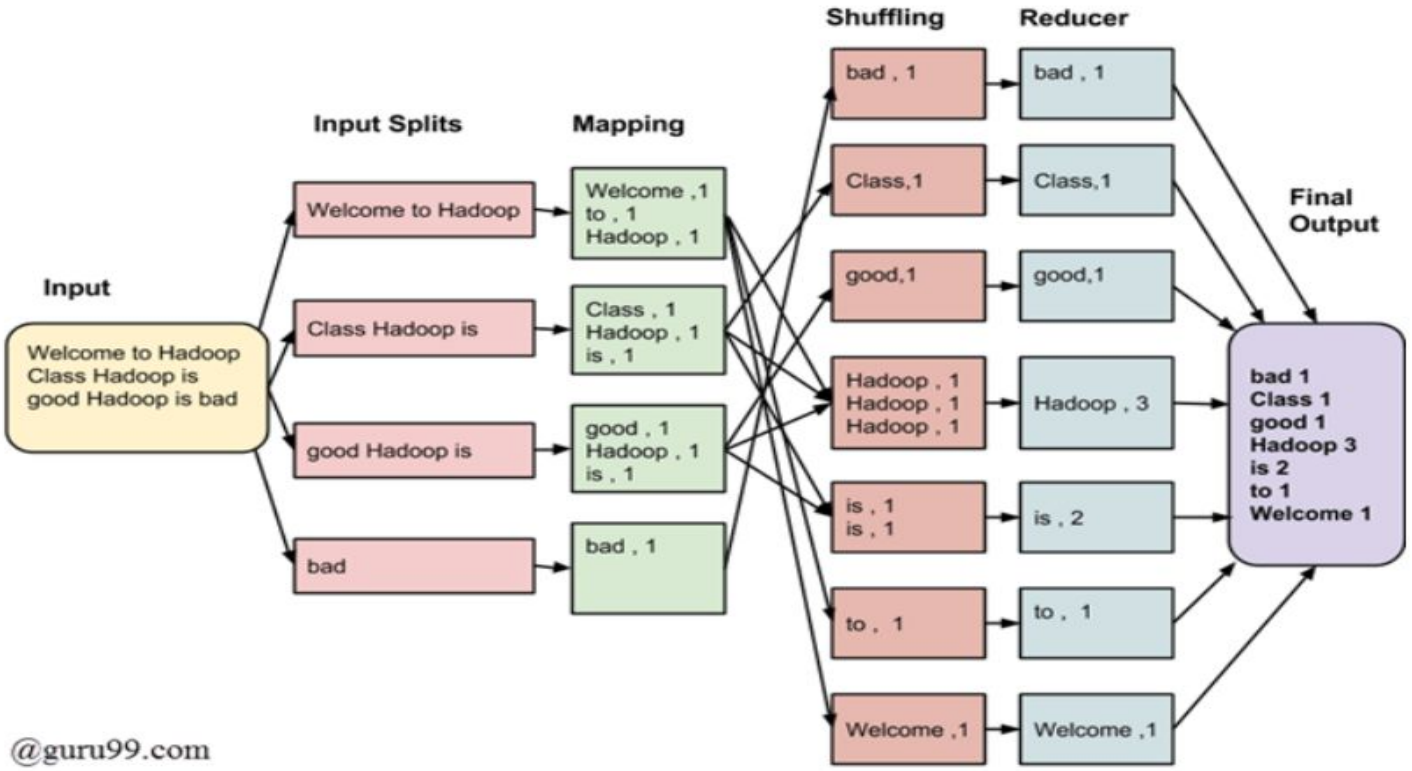
Hadoop Ecosystem - MapReduce

- The major feature of MapReduce is to perform the **distributed processing in parallel** in a Hadoop cluster which Makes Hadoop working so fast.
- MapReduce are two functions: Map and Reduce. They are sequenced one after the other.
  - **Map():** takes input from the disk as <key,value> pairs, processes them, and produces another set of intermediate <key,value> pairs as output.
    - Splitting the input and mapping is part of Map()
  - **Reduce():** also takes inputs as <key,value> pairs (output of mapping function) and produces <key,value> pairs as output.
    - Shuffling and reducing is part of Reduce()

Hadoop Ecosystem - MapReduce



Hadoop Ecosystem - MapReduce



MapReduce Architecture

Hadoop Ecosystem - Pig

- Pig is a high-level data flow platform for executing MapReduce programs of Hadoop.
- It was developed by Yahoo. The language for Pig is pig Latin.
- PIG has two parts:
  - *Pig Latin, the language*
  - *The pig runtime, for the execution environment.*
- The Pig scripts get internally converted to MapReduce jobs and get executed on data stored in HDFS.
- Apart from that, Pig can also execute its job in ***Apache Tez or Apache Spark.***
- Pig can handle any type of data, i.e., structured, semi-structured or unstructured and stores the corresponding results into Hadoop Data File System.

Hadoop Ecosystem - Pig

- Every task which can be achieved using PIG can also be achieved using java used in MapReduce.
- The compiler internally converts pig latin to MapReduce.
- It produces a sequential set of MapReduce jobs, and that’s an abstraction (which works like black box).

Hadoop Ecosystem - Hive

- HIVE is a data warehousing component which performs reading, writing and managing large data sets in a distributed environment using SQL-like interface.
- The query language of Hive is called Hive Query Language(HQL), which is very similar like SQL.
- It has 2 basic components:
  - *Hive Command Line*
  - *JDBC/ODBC driver.*
- The Hive Command line interface is used to execute HQL commands.
- Java Database Connectivity (JDBC) and Object Database Connectivity (ODBC) is used to establish connection from data storage.

Hadoop Ecosystem - Mahout

- Apache Mahout is an open source project that is primarily used for creating scalable machine learning algorithms. It implements popular machine learning techniques such as:
  - Recommendation
  - Classification
  - Clustering
- Mahout lets applications to analyze large sets of data effectively and in quick time.
- Includes several MapReduce enabled clustering implementations such as k-means, fuzzy k-means, Canopy, Dirichlet, and Mean-Shift.
- Supports Distributed Naive Bayes and Complementary Naive Bayes classification implementations.

Hadoop Ecosystem - Sparks

- Spark is an alternative framework to Hadoop built on Scala but supports varied applications written in Java, Python, etc.
- Compared to MapReduce it provides in-memory processing which accounts for faster processing.
- In addition to batch processing offered by Hadoop, it can also handle real-time processing

Hadoop Ecosystem - Hbase

- HBase is an open source, non-relational distributed database. In other words, it is a **NoSQL database.**
- It supports all types of data and that is why, it's capable of handling anything and everything inside a Hadoop ecosystem.
- It is modelled after **Google's BigTable**, which is a distributed storage system designed to cope up with large data sets.
- The **HBase was designed to run on top of HDFS and provides BigTable like capabilities.**
- It gives us a fault tolerant way of storing sparse data, which is common in most Big Data use cases.
- The HBase is written in Java, whereas HBase applications can be written in REST, Avro and Thrift APIs.

Hadoop Ecosystem - Apache Ambari

- It is an open-source tool responsible for keeping track of running applications and their statuses.
- Ambari manages, monitors, and provisions Hadoop clusters.
- Also, it also provides a central management service to start, stop, and configure Hadoop services.

Hadoop Ecosystem - Apache Flumes

- Apache Flume is an open-source tool for collecting, aggregating, and moving huge amounts of streaming data from the external web servers to the central store, say HDFS, HBase, etc.
- It is a highly available and reliable service which has tunable recovery mechanisms.
- The main purpose of designing Apache Flume is to **move streaming data** generated by various applications to **Hadoop Distributed File System.**
- Apache Flume can efficiently ingest log data from various servers into a centralized repository.
- With Flume, we can collect data from different web servers in real-time as well as in batch mode.

Hadoop Ecosystem - Sqoop

- Sqoop is a tool used to transfer bulk data between Hadoop and external datastores, such as relational databases (MS SQL Server, MySQL).
- To process data using Hadoop, the data first needs to be loaded into Hadoop clusters from several sources
- Sqoop in Hadoop helped to overcome all the challenges of the traditional approach and it could load bulk data from RDBMS to Hadoop with ease.
    - Maintaining data consistency
    - Ensuring efficient utilization of resources
    - Loading bulk data to Hadoop was not possible
    - Loading data using scripts was slow

Hadoop Ecosystem - Kafka

- There are a lot of applications generating data and a commensurate number of applications consuming that data.
- But connecting them individually is a tough task. That’s where Kafka comes in.
- It sits between the applications generating data (Producers) and the applications consuming data (Consumers).
- Kafka is distributed and has in-built partitioning, replication, and fault-tolerance.
- It can handle streaming data and also allows businesses to analyze data in real-time.

Hadoop Ecosystem - Zookeeper

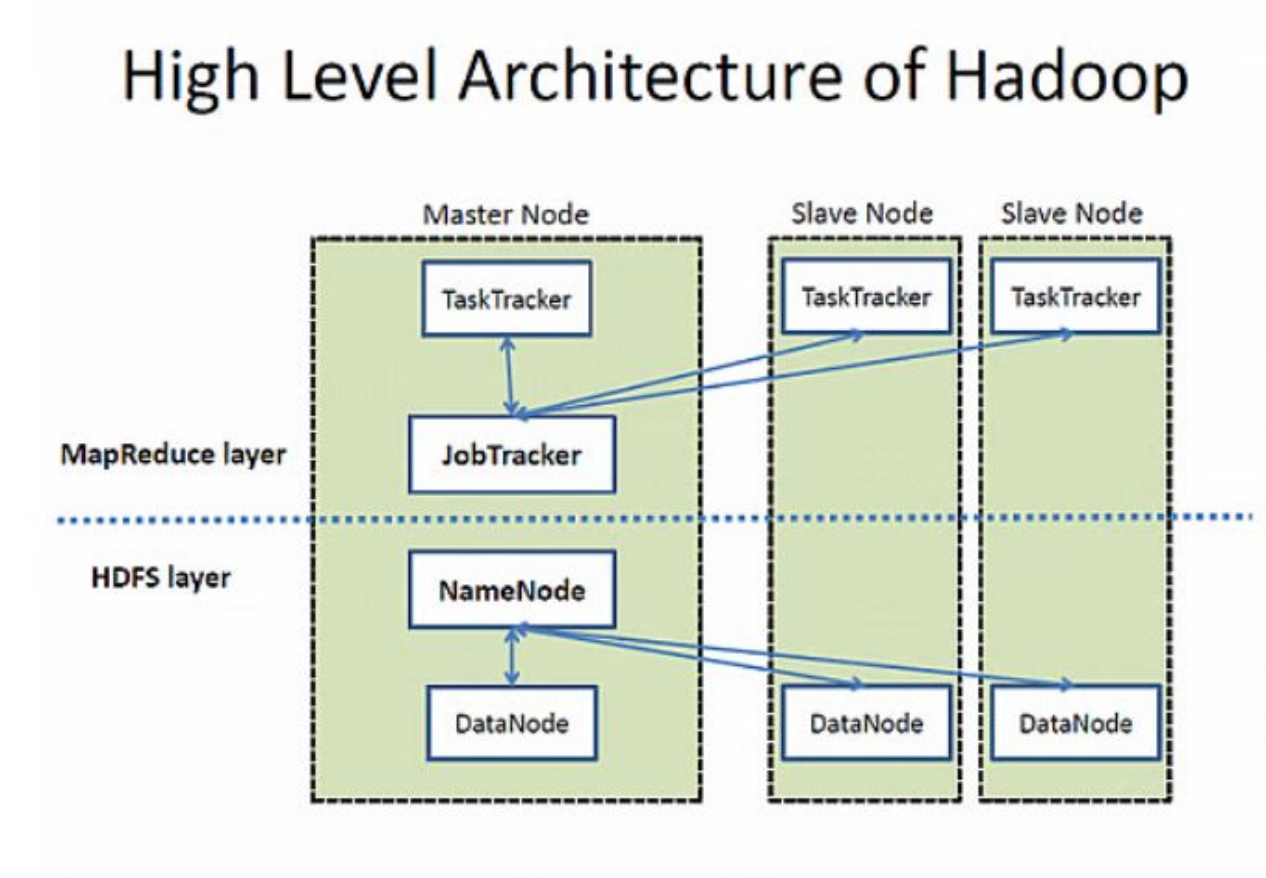
- Apache Zookeeper is the coordinator of any Hadoop job which includes a combination of various services in a Hadoop Ecosystem.
- Apache Zookeeper coordinates with various services in a distributed environment.
- It saves a lot of time by performing synchronization, configuration maintenance, grouping and naming.

Hadoop Ecosystem - Oozie

- For Apache jobs, Oozie has been just like a scheduler.
- It schedules Hadoop jobs and binds them together as one logical work.
- There are two kinds of Oozie jobs:
  - *Oozie workflow*: These are sequential set of actions to be executed. You can assume it as a relay race. Where each athlete waits for the last one to complete his part.
  - *Oozie Coordinator:* These are the Oozie jobs which are triggered when the data is made available to it.

Hadoop Architecture

Job Tracker & Task Tracker:
Job Tracker executes on a master node and Task Tracker executes on every slave node(master node also have a Task Tracker).
Each slave ties with processing (TaskTracker) and the storage (DataNode).
The Job Tracker maintains records of each resource in the Hadoop cluster. It schedules and assigns resources to the Task Tracker nodes.
It regularly receives the progress status from Task Tracker.
The task tracker regularly receives execution requests from Job Tracker. Task tracker tracks each task(mapper or reducer) and updates Job tracker about the workload.
Task-Trackers' main responsibility is to manage the processing resources on each slave node.



Hadoop Versions

END (Part 1)