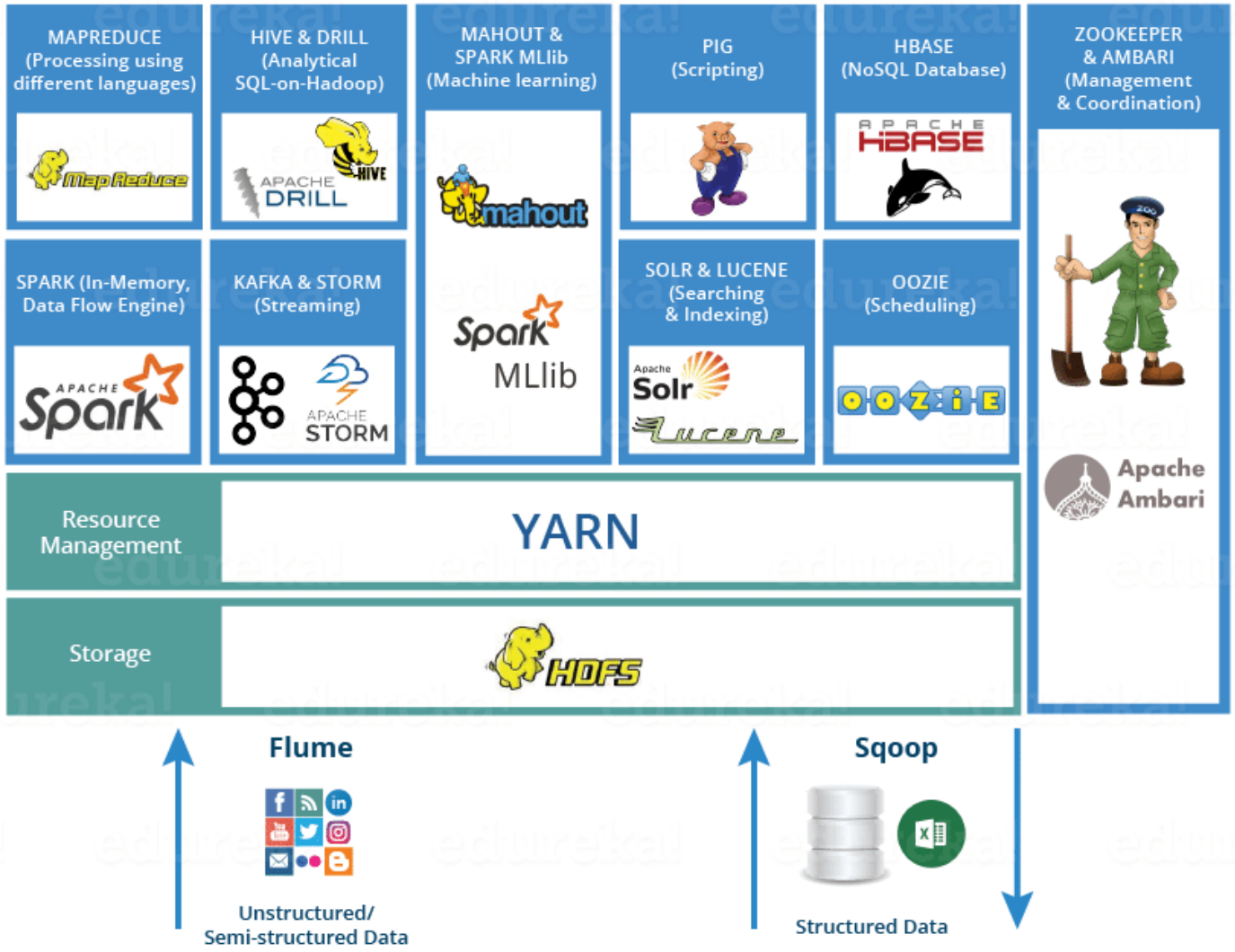
Chapter 2

Hadoop Ecosystem

Hadoop ecosystem, Core components of the Hadoop ecosystem:

Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There are four major elements of Hadoop i.e. **HDFS, MapReduce, YARN, and Hadoop Common Utilities**. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

Hadoop ecosystem, Core components of the Hadoop ecosystem:



Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **HDFS:**

- 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 and thereby maintaining the metadata in the form of log files.
- HDFS consists of two core components i.e.
  - Name node
  - Data Node
- Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment. Undoubtedly, making Hadoop cost effective.
- HDFS maintains all the coordination between the clusters and hardware, thus working at the heart of the system.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **YARN:**

- Yet Another Resource Negotiator, 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.
- Consists of three major components i.e.
  - Resource Manager
  - Nodes Manager
  - Application Manager
- Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **MapReduce:**

- By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing's logic and helps to write applications which transform big data sets into a manageable one.
- MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:
  - *Map()* performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
  - *Reduce()*, as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **PIG:**
- Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.
- It is a platform for structuring the data flow, processing and analyzing huge data sets.
- 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.
- Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the JVM.
- Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **HIVE:**

- With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
- It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
- Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
- JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **Mahout:**

- Mahout, allows Machine Learnability to a system or application. Machine Learning, as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.
- It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **Apache Spark:**

- It’s a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.
- It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
- Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **Apache HBase:**

- It's a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google's BigTable, thus able to work on Big Data sets effectively.
- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data

Hadoop ecosystem, Core components of the Hadoop ecosystem:

- **Other Components:** Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:
- **Zookeeper:** There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often. Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.
- **Oozie:** Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

Reference: https://www.geeksforgeeks.org/hadoop-ecosystem/

Hadoop master/slave architecture: (whenever Hadoop master slave architecture is asked, you need to write both HDFS master/slave and Map-Reduce (YARN) master/slave architecture):

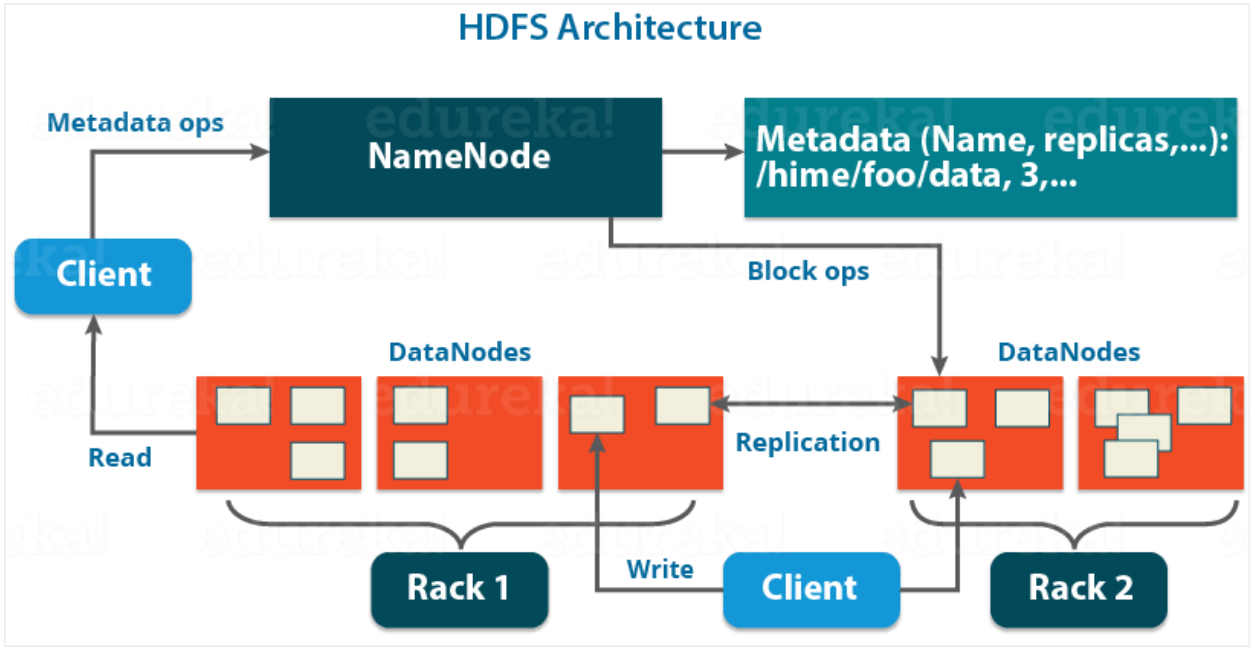
Hadoop master/slave architecture:

- Hadoop is a framework written in Java that utilizes a large cluster of commodity hardware to maintain and store big size data. Hadoop works on MapReduce Programming Algorithm that was introduced by Google. Today lots of Big Brand Companies are using Hadoop in their Organization to deal with big data, eg. Facebook, Yahoo, Netflix, eBay, etc. The Hadoop Architecture Mainly consists of 4 components.
- MapReduce
- HDFS(Hadoop Distributed File System)
- YARN(Yet Another Resource Negotiator)
- Common Utilities or Hadoop Common

Hadoop master/slave architecture:

Apache HDFS or **Hadoop Distributed File System** 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 can be deployed on a broad spectrum of machines that support Java. Though one can run several DataNodes on a single machine, but in the practical world, these DataNodes are spread across various machines.

Hadoop master/slave architecture:



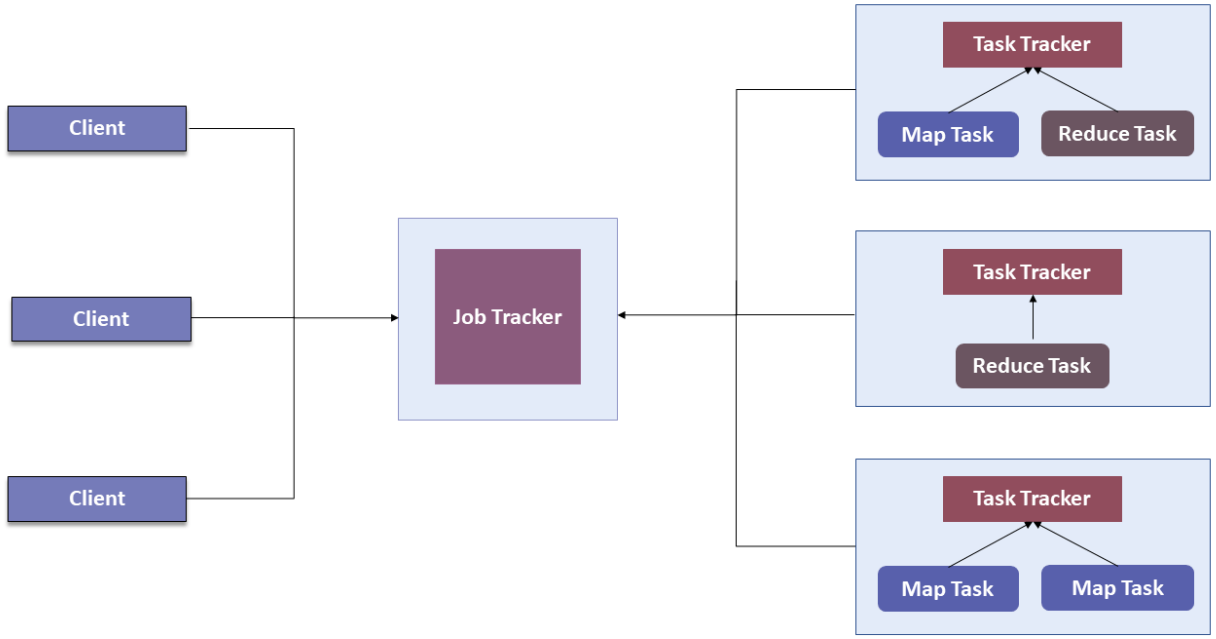
Hadoop master/slave architecture:

- NameNode is the master node in the Apache Hadoop HDFS Architecture that maintains and manages the blocks present on the DataNodes (slave nodes). NameNode is a very highly available server that manages the File System Namespace and controls access to files by clients. The HDFS architecture is built in such a way that the user data never resides on the NameNode. The data resides on DataNodes only.
- DataNodes are the slave nodes in HDFS. Unlike NameNode, DataNode is a commodity hardware, that is, a non-expensive system which is not of high quality or high-availability. The DataNode is a block server that stores the data in the local file ext3 or ext4.
- Apart from these two daemons, there is a third daemon or a process called Secondary NameNode. The Secondary NameNode works concurrently with the primary NameNode as a **helper daemon.** The secondary Namenode is not backup for Namenode

Hadoop master/slave architecture:

In Hadoop version 1.0 which is also referred to as MRV1(MapReduce Version 1), MapReduce performed both processing and resource management functions. It consisted of a Job Tracker which was the single master. The Job Tracker allocated the resources, performed scheduling and monitored the processing jobs. It assigned map and reduce tasks on a number of subordinate processes called the Task Trackers. The Task Trackers periodically reported their progress to the Job Tracker.

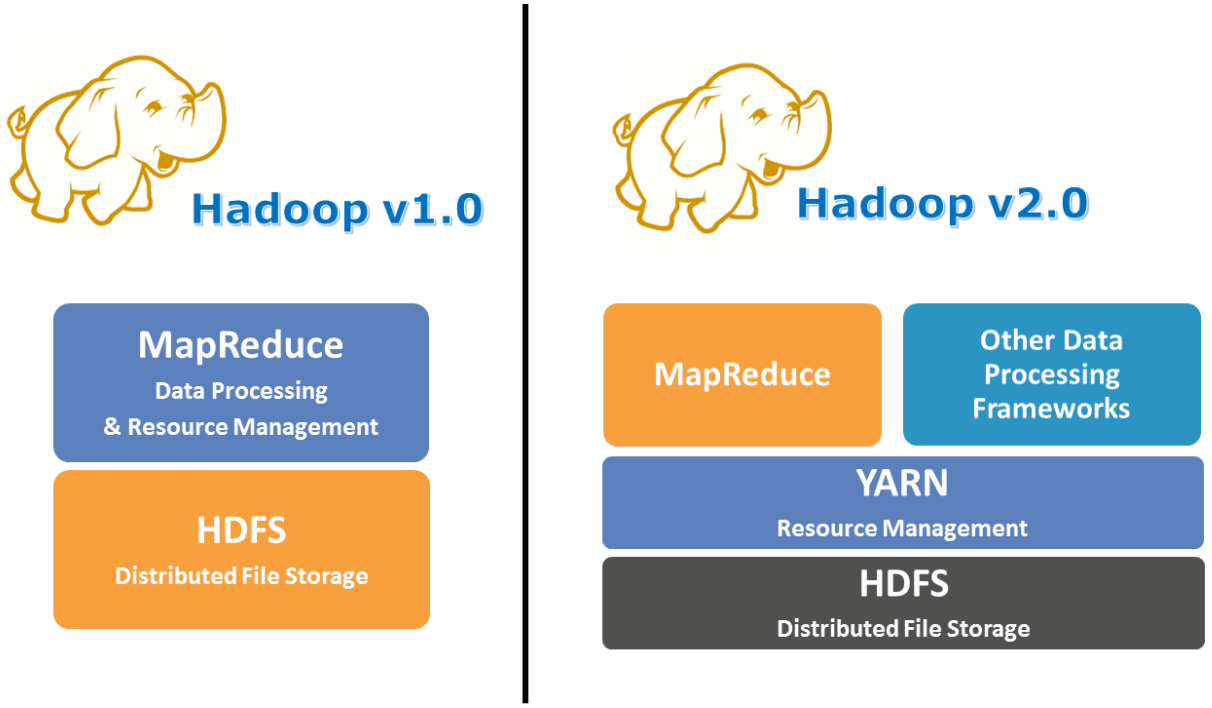
Hadoop master/slave architecture:



Hadoop master/slave architecture:

- This design resulted in scalability bottleneck due to a single Job Tracker. IBM mentioned in its article that according to Yahoo!, the practical limits of such a design are reached with a cluster of 5000 nodes and 40,000 tasks running concurrently. Apart from this limitation, the utilization of computational resources is inefficient in MRV1. Also, the Hadoop framework became limited only to MapReduce processing paradigm.
- To overcome all these issues, YARN was introduced in Hadoop version 2.0 in the year 2012 by Yahoo and Hortonworks. The basic idea behind YARN is to relieve MapReduce by taking over the responsibility of Resource Management and Job Scheduling. YARN started to give Hadoop the ability to run non-MapReduce jobs within the Hadoop framework.
- YARN allows different data processing methods like graph processing, interactive processing, stream processing as well as batch processing to run and process data stored in HDFS. Therefore YARN opens up Hadoop to other types of distributed applications beyond MapReduce.

Hadoop master/slave architecture:

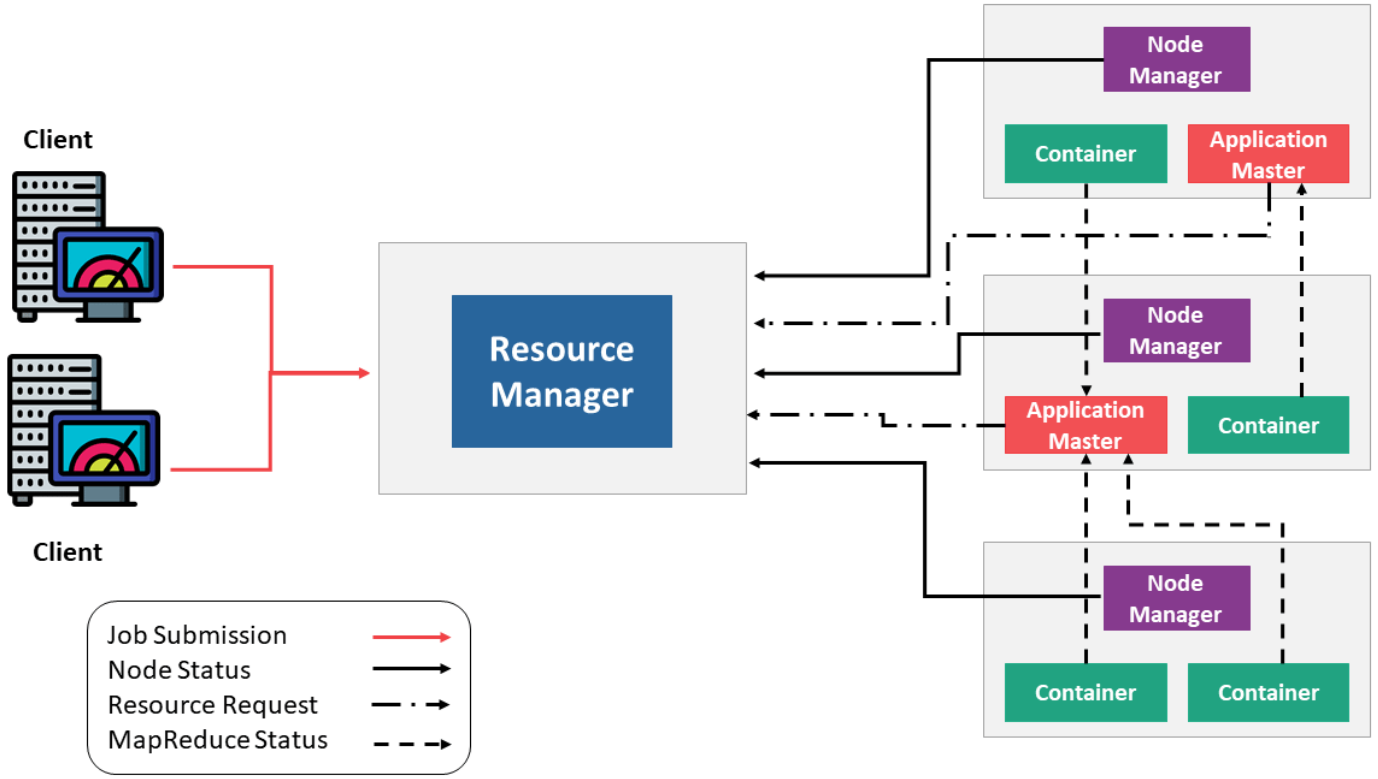


Hadoop master/slave architecture:

Apart from Resource Management, YARN also performs Job Scheduling. YARN performs all your processing activities by allocating resources and scheduling tasks. Apache Hadoop YARN Architecture consists of the following main components :

- **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

Hadoop master/slave architecture:



Hadoop master/slave architecture:

- The **first component** of YARN Architecture is 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.
- It has two major components: a) Scheduler b) Application Manager

Hadoop master/slave architecture:

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.

Hadoop master/slave architecture:

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.

Hadoop master/slave architecture:

The **third component** of Apache Hadoop YARN is Application Master

Application Master:

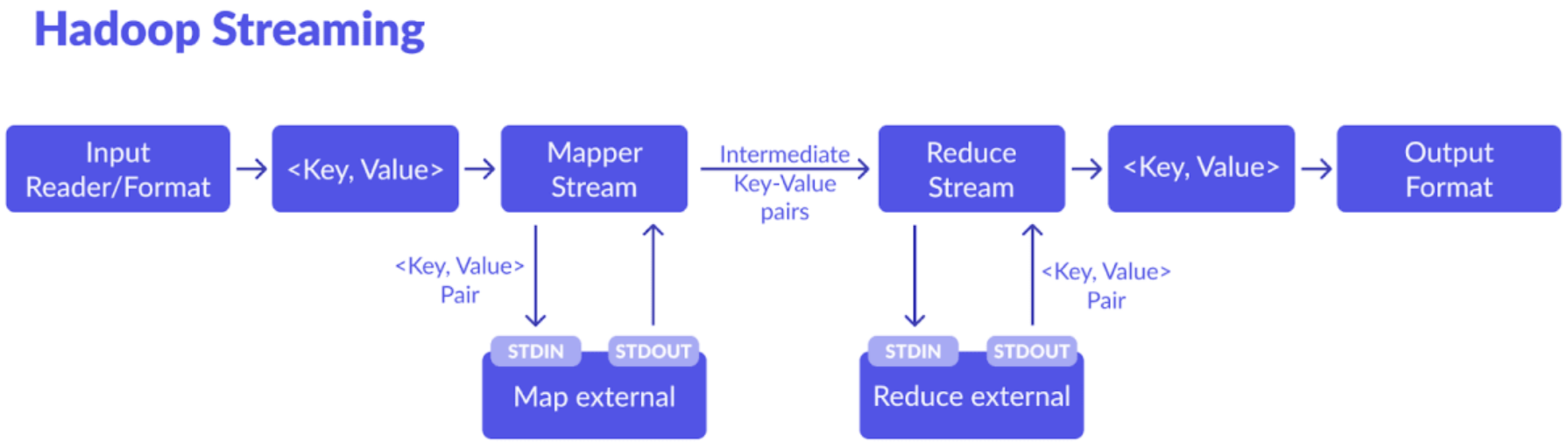
- An application is a single job submitted to the framework. Each such application has a unique Application Master associated with it which is a framework specific entity.
- It is the process that coordinates an application's execution in the cluster and also manages faults.
- Its task is to negotiate 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.

Hadoop daemons, Hadoop configuration modes: Refer to the chapter 2.pdf

Hadoop streaming: (short notes only)

- It is a utility or feature that comes with a Hadoop distribution that allows developers or programmers to write the Map-Reduce program using different programming languages like Ruby, Perl, Python, C++, etc. We can use any language that can read from the standard input(STDIN) like keyboard input and all and write using standard output(STDOUT). We all know the Hadoop Framework is completely written in java but programs for Hadoop are not necessarily need to code in Java programming language. Feature of Hadoop Streaming is available since Hadoop version 0.14.1.

Hadoop streaming: (short notes only)



Hadoop streaming: (short notes only)

In the diagram above, the Mapper reads the input data from Input Reader/Format in the form of key-value pair, maps them as per logic written on code, and then passes through the Reduce stream, which performs data aggregation and releases the data to the output

Hadoop streaming: (short notes only)

- Hadoop Streaming provides several important features:
- Users can execute non-Java-programmed MapReduce jobs on Hadoop clusters. Supported languages include Python, Perl, and C++.
- Hadoop Streaming monitors the progress of jobs and provides logs of a job's entire execution for analysis.
- Hadoop Streaming works on the MapReduce paradigm, so it supports scalability, flexibility, and security/authentication.
- Hadoop Streaming jobs are quick to develop and don't require much programming (except for executables).

Important Questions from Chapter 2.

- Explain why Hadoop fails when master node fails.
- What changes were made to Hadoop after Version 2.
- Explain about Hadoop configuration modes (see chapter2.pdf)
- Explain about different daemons of Hadoop. (see chapter2.pdf)
- What does running Hadoop means? (It is asking about daemons)
- What is the advantage of using master slave architecture in Hadoop?
- Ecosystem of Hadoop
- Core component of Hadoop: HDFS, YARN, Map Reduce, Hadoop Common