Introduction to Hadoop and MapReduce Programming

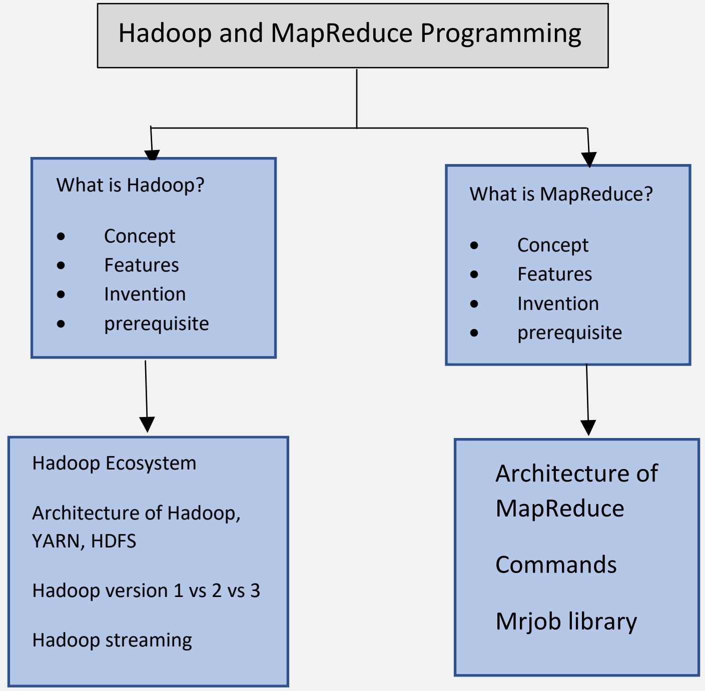
Hadoop is an open-source framework from Apache and is used to store process and analyse data which are very huge in volume. And a MapReduce is a data processing tool which is used to process the data parallelly in a distributed form.

As a part of Hadoop and MapReduce Programming, you covered:

- Introduction to Hadoop
- Hadoop deep drive
- Introduction to MapReduce
- MapReduce Architecture, Commands and Libraries

Common Interview Questions:

1. What platform and Java version are required to run Hadoop?
2. What are the most common input formats defined in Hadoop?
3. What is JobTracker and NameNode in Hadoop?
4. What are the functionalities of Jobtracker?
5. What is shuffling in MapReduce?
6. What is "map" and what is "reducer" in Hadoop?
7. What is heartbeat in HDFS?
8. What happens when a data node fails?
9. What are the network requirements for using Hadoop?



Hadoop and MapReduce Programming

What is Hadoop?

- Concept
- Features
- Invention
- prerequisite

What is MapReduce?

- Concept
- Features
- Invention
- prerequisite

Hadoop Ecosystem

Architecture of Hadoop, YARN, HDFS

Hadoop version 1 vs 2 vs 3

Hadoop streaming

Architecture of MapReduce

Commands

Mrjob library

Introduction to Hadoop and MapReduce Programming

Hadoop:

Hadoop is an open-source framework from Apache and is used to store process and analyze data which are very huge in volume.

The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers.

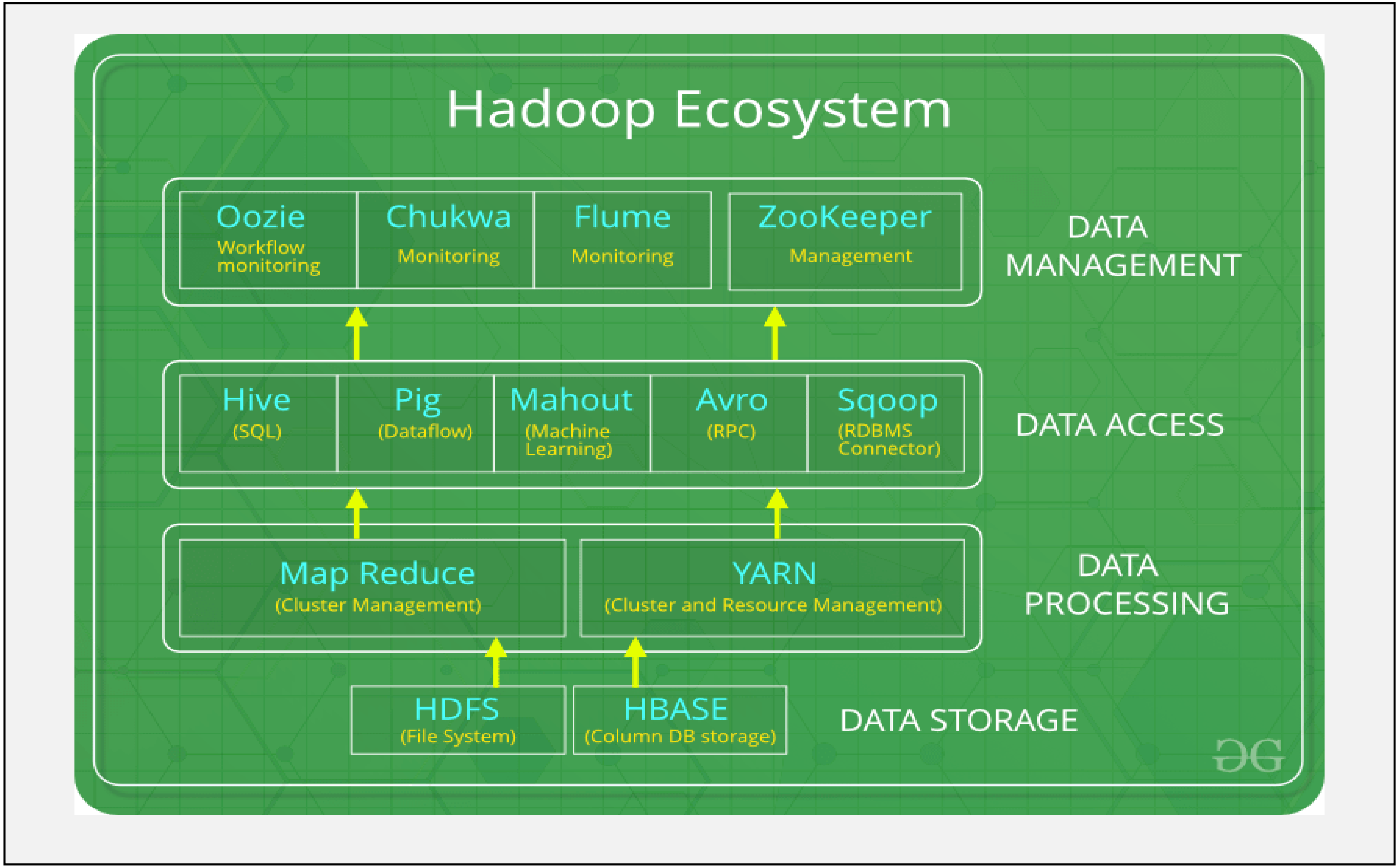
MapReduce:

A MapReduce is a data processing tool which is used to process the data parallelly in a distributed form.

The MapReduce algorithm contains two important tasks, namely Map and Reduce

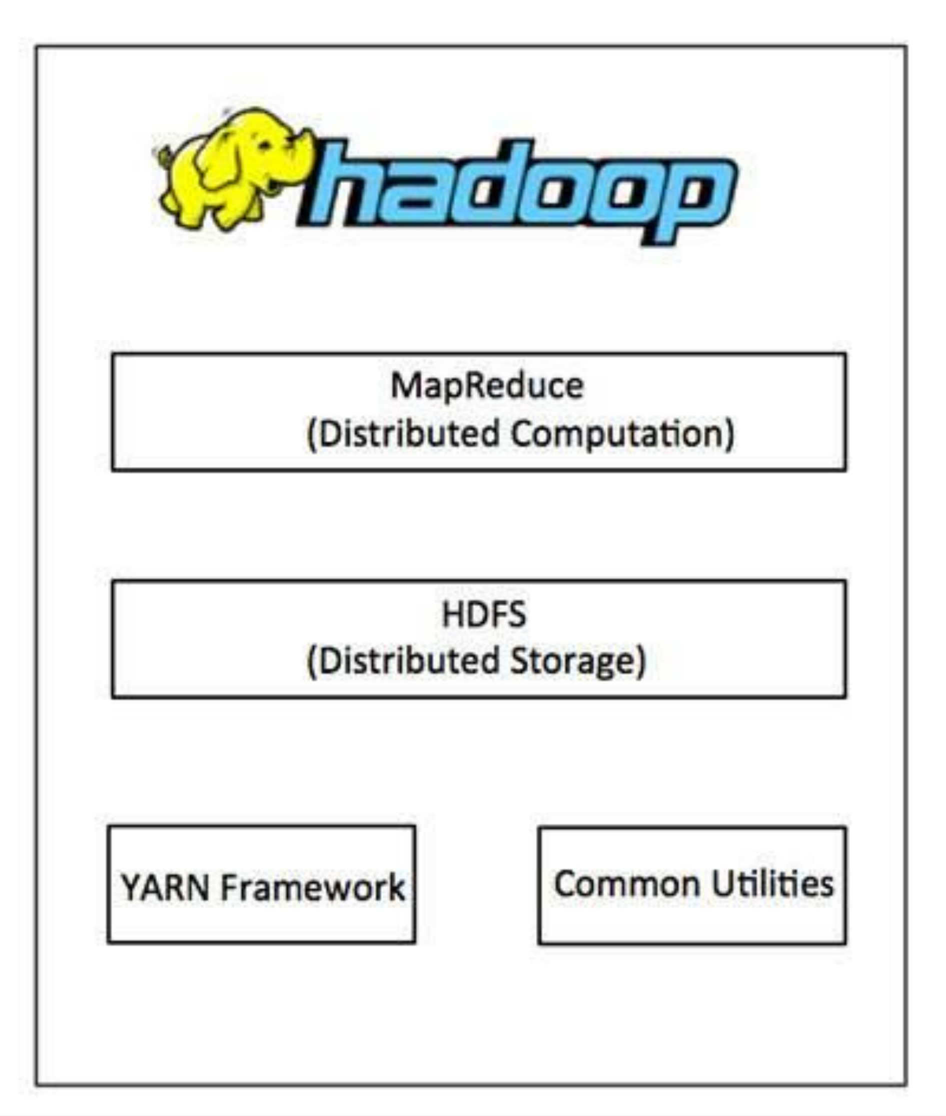
Based on	Hadoop	MapReduce
Concept	The Apache Hadoop is an eco-system which provides an environment which is reliable, scalable and ready for distributed computing.	MapReduce is a submodule of this project which is a programming model and is used to process huge datasets which sits on HDFS.
Features	Hadoop is Open Source Hadoop cluster is Highly Scalable	MapReduce provides Fault Tolerance MapReduce provides High Availability
Invention	Hadoop was created by Doug Cutting and Mike Cafarella.	MapReduce is invented by Google
Pre-requisites	Hadoop runs on HDFS (Hadoop Distributed File System)	MapReduce can run on HDFS/GFS/NDFS or any other distributed system for example MapR-FS

Hadoop Ecosystem:



Introduction to Hadoop and MapReduce Programming

Hadoop Architecture:



Important Terminologies:

Mapper − Mapper maps the input key/value pairs to a set of intermediate key/value pair.

NamedNode − Node that manages the Hadoop Distributed File System (HDFS).

DataNode − Node where data is presented in advance before any processing takes place.

MasterNode − Node where JobTracker runs and which accepts job requests from clients.

SlaveNode − Node where Map and Reduce program runs.

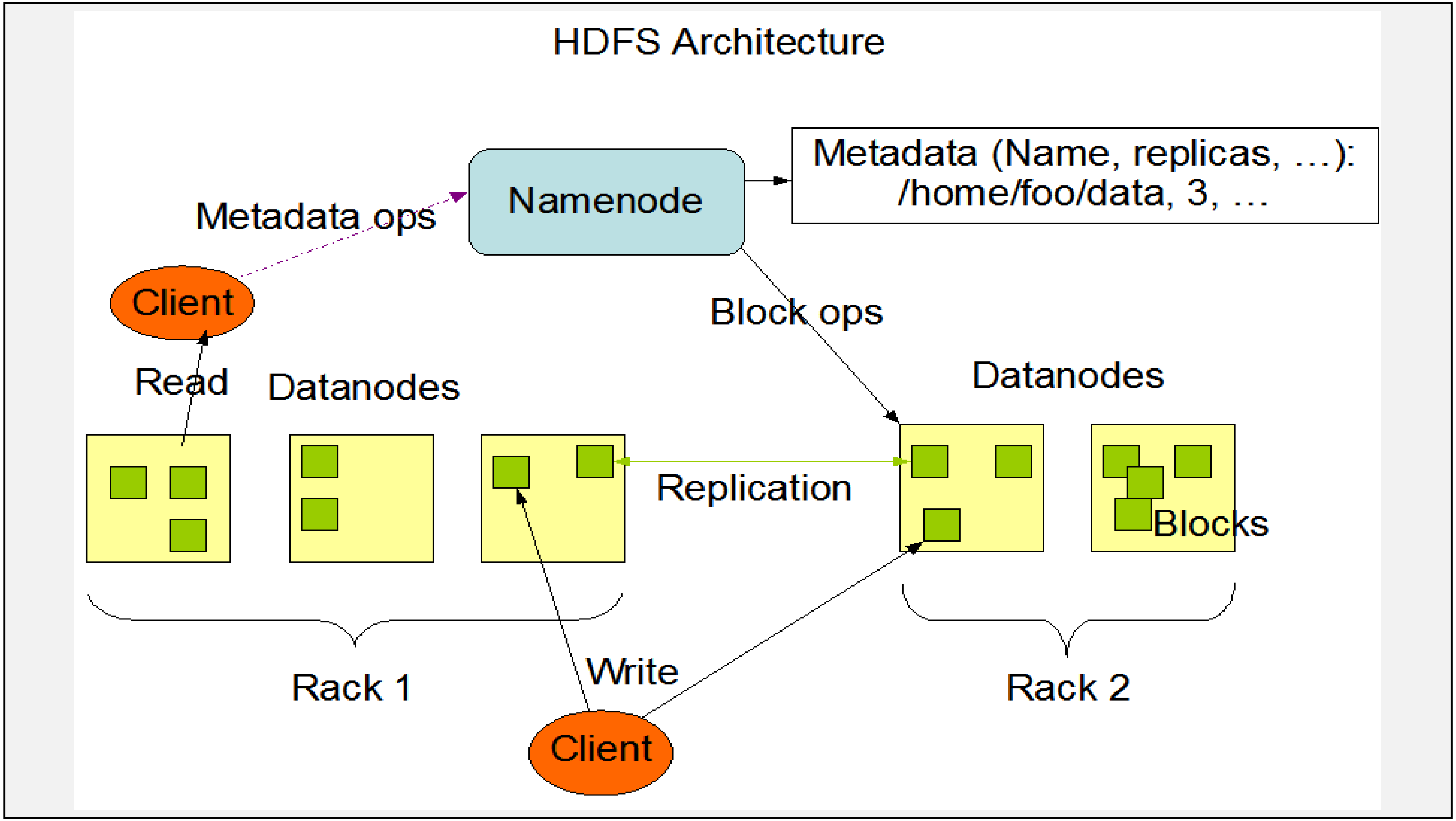
JobTracker − Schedules jobs and tracks the assign jobs to Task tracker.

Task Tracker − Tracks the task and reports status to JobTracker.

Job − A program is an execution of a Mapper and Reducer across a dataset.

Task Attempt − A particular instance of an attempt to execute a task on a SlaveNode.

HDFS Architecture:



Introduction to Hadoop and MapReduce Programming

YARN:

YARN stands for "Yet Another Resource Negotiator". It was introduced in Hadoop 2.0 to remove the bottleneck on Job Tracker which was present in Hadoop 1.0. YARN was described as a "Redesigned Resource Manager" at the time of its launching, but it has now evolved to be known as large-scale distributed operating system used for Big Data processing.

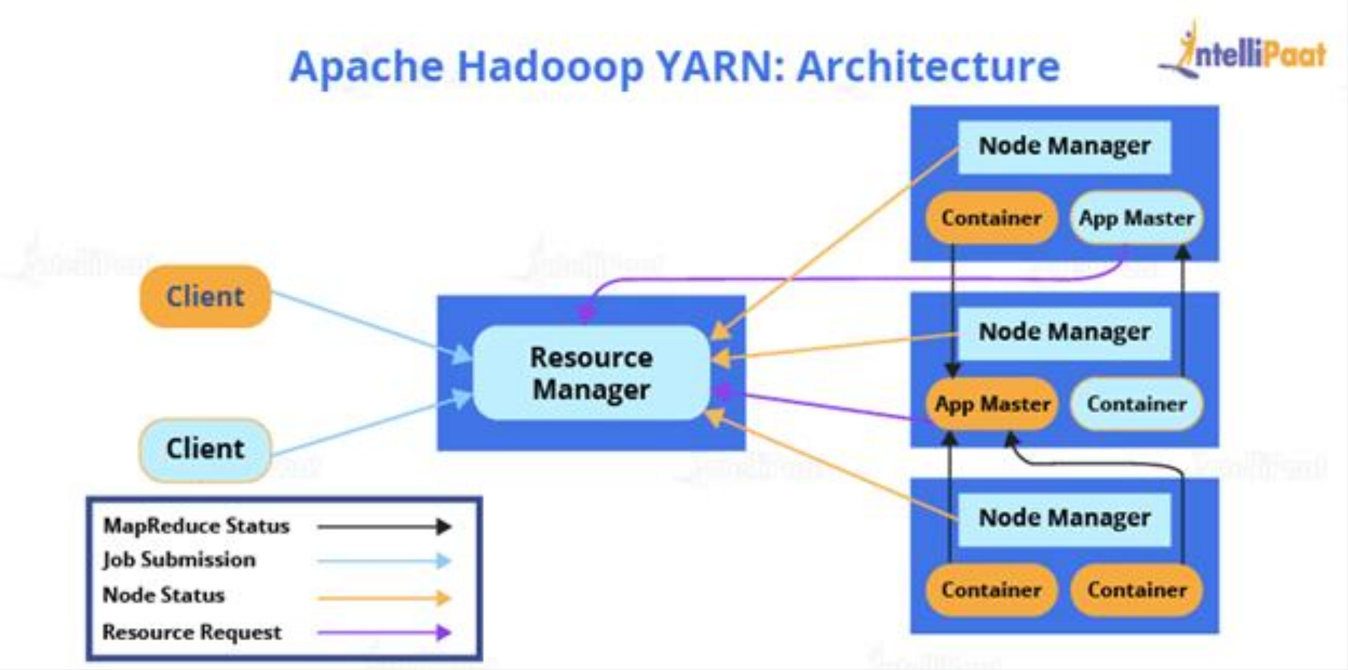
YARN Components:

Client: For submitting MapReduce jobs.

Resource Manager: To manage the use of resources across the cluster

Node Manager:For launching and monitoring the computer containers on machines in the cluster.

Map Reduce Application Master: Checks tasks running the MapReduce job



Difference between Hadoop version 1, 2 and 3:

Hadoop 1	Hadoop 2
Limited to 4000 nodes, per Cluster	Up to 10,000 nodes per cluster
Hadoop 1 has some less components and APIs as compare to that of Hadoop 2.	Hadoop 2 has more components and APIs such as YARN API, YARN FRAMEWORK, and enhanced Resource Manager.
Has only one name space for handling HDFS	Supports multiple name space for handling HDFS.
‘0’ number of tasks in a cluster	‘0’ (Cluster Size)
Jobtracker bottleneck	Efficient cluster utilization-YARN
Map and reduce slots are static	Not restricted to Java
Has only one job- to run MapReduce	Any Application can Integrate with hadoop

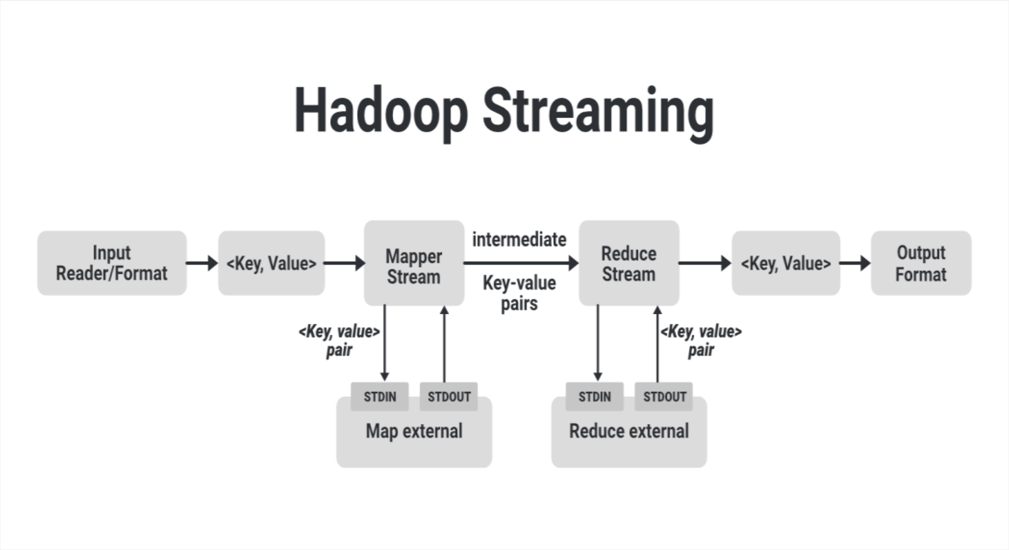
Hadoop 2	Hadoop 3
Handling Fault tolerance through replication	Handling Fault tolerance through erasure coding
Consumes 200% in HDFS Storage	Consumes just 50% storage
Up to 10,000 nodes per cluster	Over 10,000 nodes per cluster
File system – DFS, FTP and Amazon S3	All features plus Microsoft Azure data lake file system
Cluster resource management is handled by YARN	Cluster resource management is also handled by YARN
Uses HDFS Balancer for this purpose	Uses intra-data node balancer
Minimum supported JAVA version 7	Minimum supported JAVA version 8
YARN Timeline service is not scalable beyond small clusters.	YARN Timeline service highly scalable and reliable.

Introduction to Hadoop and MapReduce Programming

Hadoop Streaming:

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).

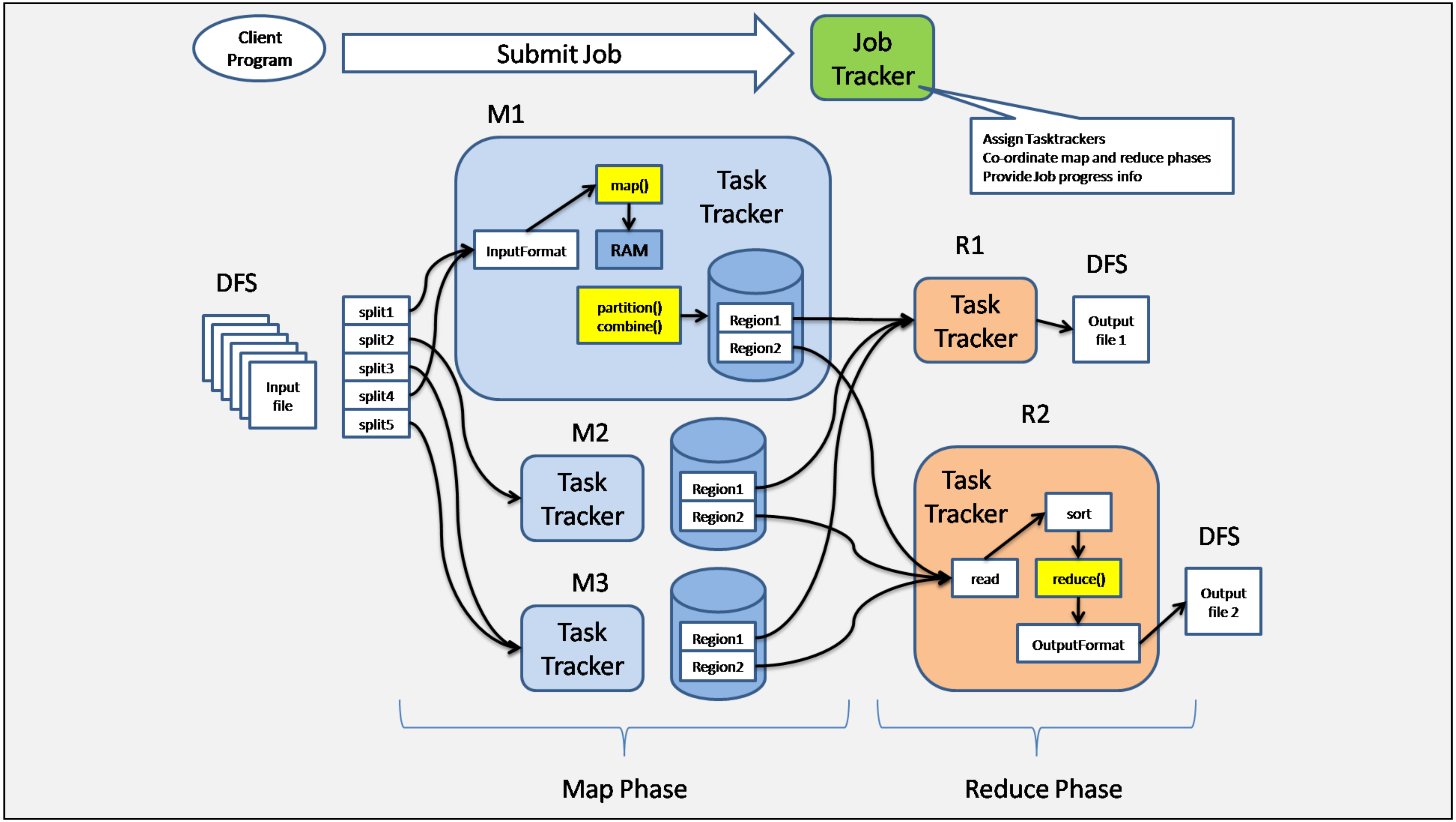
How Hadoop streaming works:



Some Hadoop streaming commands:

SR No.	GENERIC_OPTION & Description
-input directory_name or filename	Input location for the mapper.
-output directory_name	Input location for the reducer.
-mapper executable or JavaClassName	The command to be run as the mapper
-reducer executable or script or JavaClassName	The command to be run as the reducer
-file file-name	Make the mapper, reducer, or combiner executable available locally on the compute nodes
-partitioner JavaClassName	The Class that determines which key to reduce.
-combiner *streamingCommand or JavaClassName*	The Combiner executable for map output

MapReduce Architecture:



Introduction to Hadoop and MapReduce Programming

How to Interact with MapReduce Jobs.

Usage − **hadoop job [GENERIC_OPTIONS]**. The following are the Generic Options available in a Hadoop job.

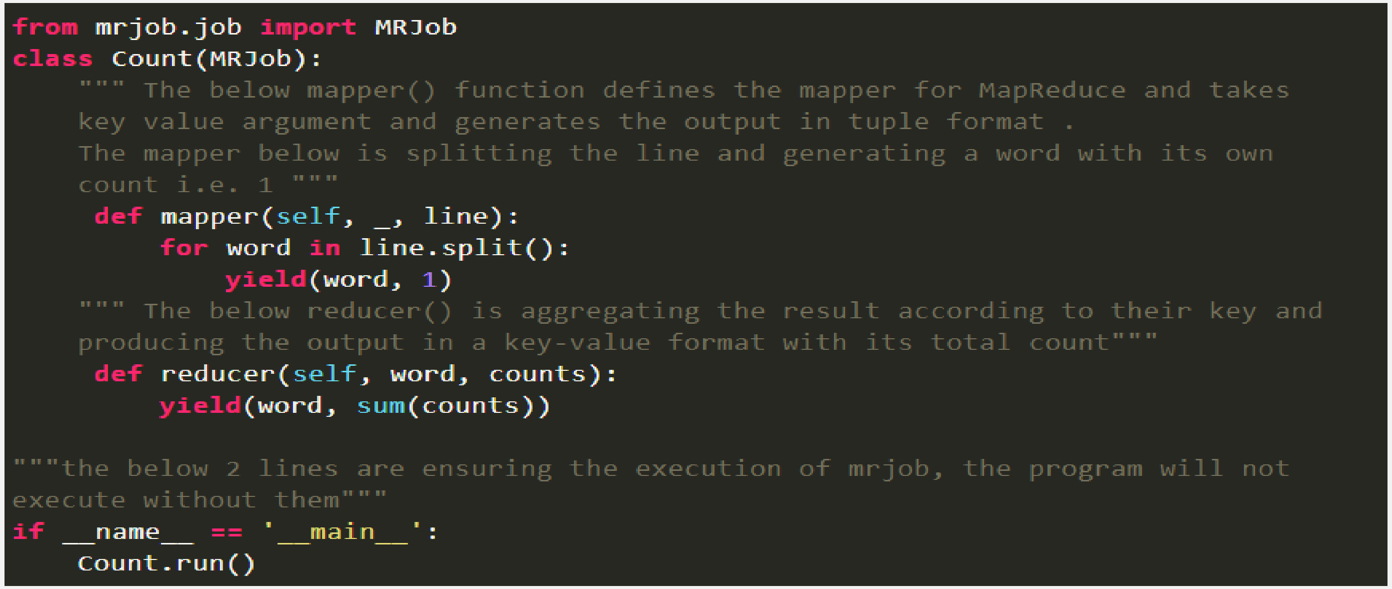
SR No.	GENERIC_OPTION & Description
1	**-submit <job-file>** Submits the job.
2	**-status <job-id>** Prints the map and reduce completion percentage and all job counters
3	**-counter <job-id> <group-name> <countername>** Prints the counter value.
4	**-kill <job-id>** Kills the job.
5	**-events <job-id> <fromevent-#> <#-of-events>** Prints the events' details received by jobtracker for the given range.
6	**-history [all] <jobOutputDir> - history < jobOutputDir>** Prints job details, failed and killed tip details
7	**-list[all]** Displays all jobs. -list displays only jobs which are yet to complete.
8	**-kill-task <task-id>** Kills the task. Killed tasks are NOT counted against failed attempts.
9	**-fail-task <task-id>** Fails the task. Failed tasks are counted against failed attempts.
10	**-set-priority <job-id> <priority>** Changes the priority of the job. Allowed priority values are VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW

mrjob Library:

mrjob is the famous python library for MapReduce developed by YELP.

- The library helps developers to write MapReduce code using a Python Programming language. Developers can test the MapReduce Python code written with mrjob locally on their system or on the cloud using Amazon EMR(Elastic MapReduce).
- mrjob is currently an active Framework for MapReduce programming or Hadoop Streaming jobs and has good document support for Hadoop with python than any other library or framework currently available.
- With mrjob, we can write code for Mapper and Reducer in a single class.
- mrjob supports Python 2.7/3.4+.

Example: Count the number of occurrence of words from a text file using python mrjob



```
from mrjob.job import MRJob
class Count(MRJob):
    """ The below mapper() function defines the mapper for MapReduce and takes
    key value argument and generates the output in tuple format .
    The mapper below is splitting the line and generating a word with its own
    count i.e. 1 """
     def mapper(self, _, line):
         for word in line.split():
             yield(word, 1)
    """ The below reducer() is aggregating the result according to their key and
    producing the output in a key-value format with its total count"""
     def reducer(self, word, counts):
         yield(word, sum(counts))

"""the below 2 lines are ensuring the execution of mrjob, the program will not
execute without them"""
if __name__ == '__main__':
    Count.run()
```