Introduction to HADOOP and HADOOP Architecture

(Unit-III)
Part -II

Dr. V. K. Patle
Assistant Professor
S. o. S. Computer Science & IT
Pt. Ravishankar Shukla University, Raipur (C.G.)

Open Source Big Data Tools

What does *Open-Source Tools* mean?

Open-source tools are software tools that are freely available without a commercial license. Many different kinds of open-source tools allow developers and others to do certain things in programming, maintaining technologies or other types of technology tasks.

Open-source tools stand in contrast to tools that are commercially licensed and available to users for a free. Well-known examples of open-source tools include many of the software products from the Apache Foundation, such as big-data tool HADOOP and related tools. Most of these are freely available, with the licensing held by a user community, instead of a company making a profit from software.

Open Source Big Data Tools

- **Apache Kafka:** It allows users to publish and subscribe to real-time data feeds. It aims to bring the reliability of other messaging systems to streaming data.
- **Apache Lucene:** It is a full-text indexing and search software library that can be used for recommendation engines. It's also the basis for many other search projects, including Solr and Elastic search.
- **Apache Pig:** It is a platform for analyzing large data sets that run on Hadoop. Yahoo, which developed it to do MapReduce jobs on large data sets, contributed it to the ASF in 2007.
- **Apache Solr:** It is an enterprise search platform built upon Lucene.
- **Apache Zeppelin:** It is an incubating project that enables interactive data analytics with SQL and other programming languages.

Other open source big data tools you may want to investigate include:

- **Elastic search:** It is another enterprise search engine based on Lucene. It's part of the Elastic stack (formerly known as the ELK stack for its components: Elastic search, Kibana, and Logstash) that generate insights from structured and unstructured data.
- **Cruise Control:** It was developed by LinkedIn to run Apache Kafka clusters at large scale.
- **TensorFlow:** It is a software library for machine learning that has grown rapidly since Google open sourced it in late 2015. It's been praised for "democratizing" machine learning because of its ease-of-use.

Apache Hadoop framework

✓Apache Hadoop is a freely licensed software framework developed by the Apache Software Foundation and used to develop data-intensive, distributed computing. Hadoop is designed to scale from a single machine up to thousands of computers. A central Hadoop concept is that errors are handled at the application layer, versus depending on hardware for reliability.

✓Hadoop was inspired by Google MapReduce and Google File System papers. It is a project developed with top-level specifications, and a community of programmers worldwide contributed to the program using Java language. Yahoo Inc. has been a major Hadoop supporter.

✓Doug Cutting is credited with the creation of Hadoop, which he named for his son’s toy elephant. Cutting’s original goal was to support the expansion of the Apache Nutch search engine project.

Apache Hadoop framework

The Apache Hadoop framework is composed of the following modules:

- **Hadoop Common:** It contains libraries and utilities needed by other Hadoop modules.
- **Hadoop Distributed File System (HDFS):** It is a distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster.
- **Hadoop YARN:** A resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- **Hadoop MapReduce:** A programming model for large scale data processing.

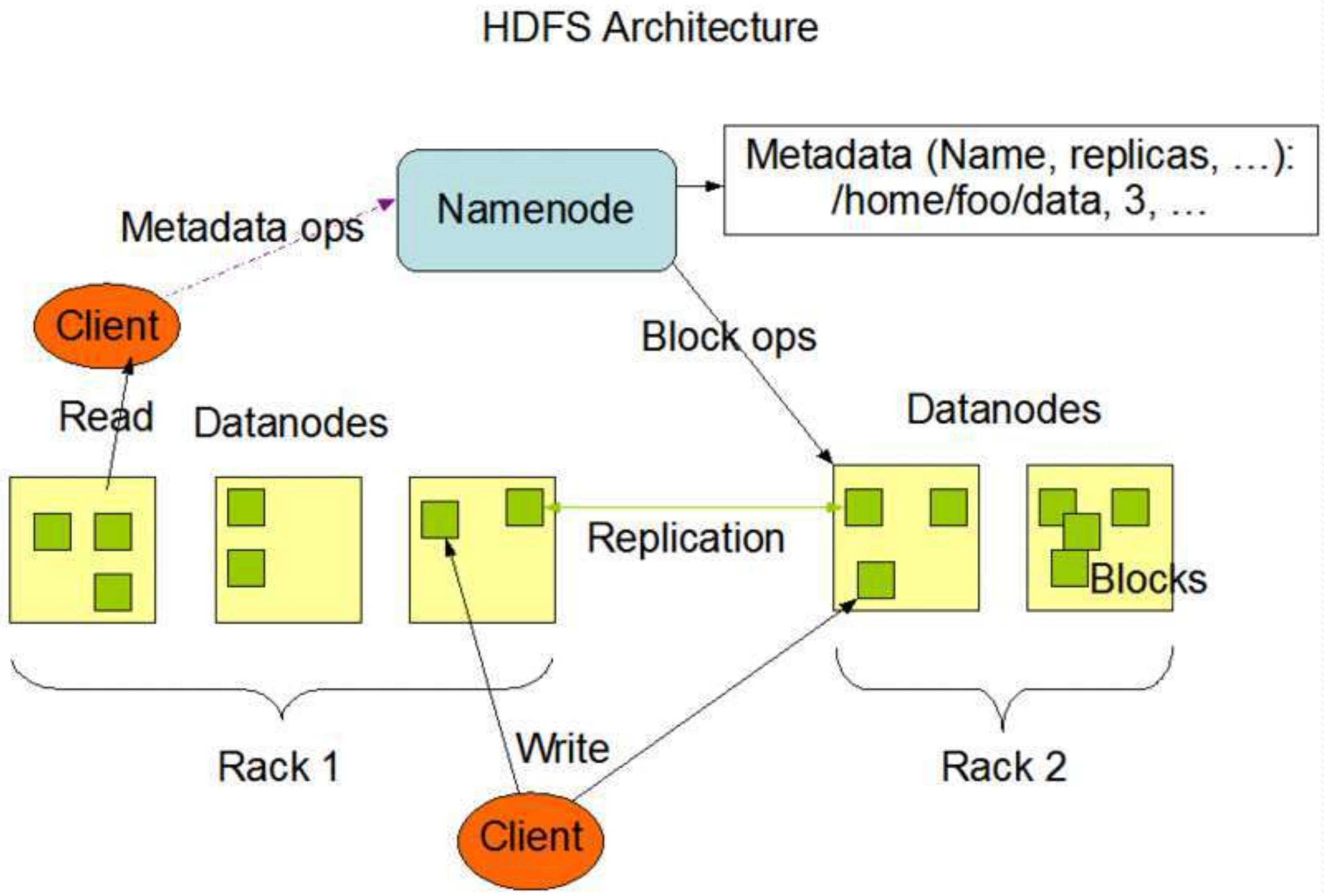
HDFS and MapReduce

There are two primary components at the core of Apache Hadoop 1.x: the Hadoop Distributed File System (HDFS) and the MapReduce parallel processing framework. These are both open source projects, inspired by technologies created inside Google.

What is HDFS ?

HDFS stands for **Hadoop Distributed File System**. It is a distributed file system of Hadoop to run on large clusters reliably and efficiently. Also, it is based on the Google File System (GFS). Moreover, it also has a list of commands to interact with the file system.

Furthermore, the HDFS works according to the master, slave architecture. The master node or name node manages the file system metadata while the slave nodes or the data notes store actual data.

HDFS Architecture
Metadata ops
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

Cont……

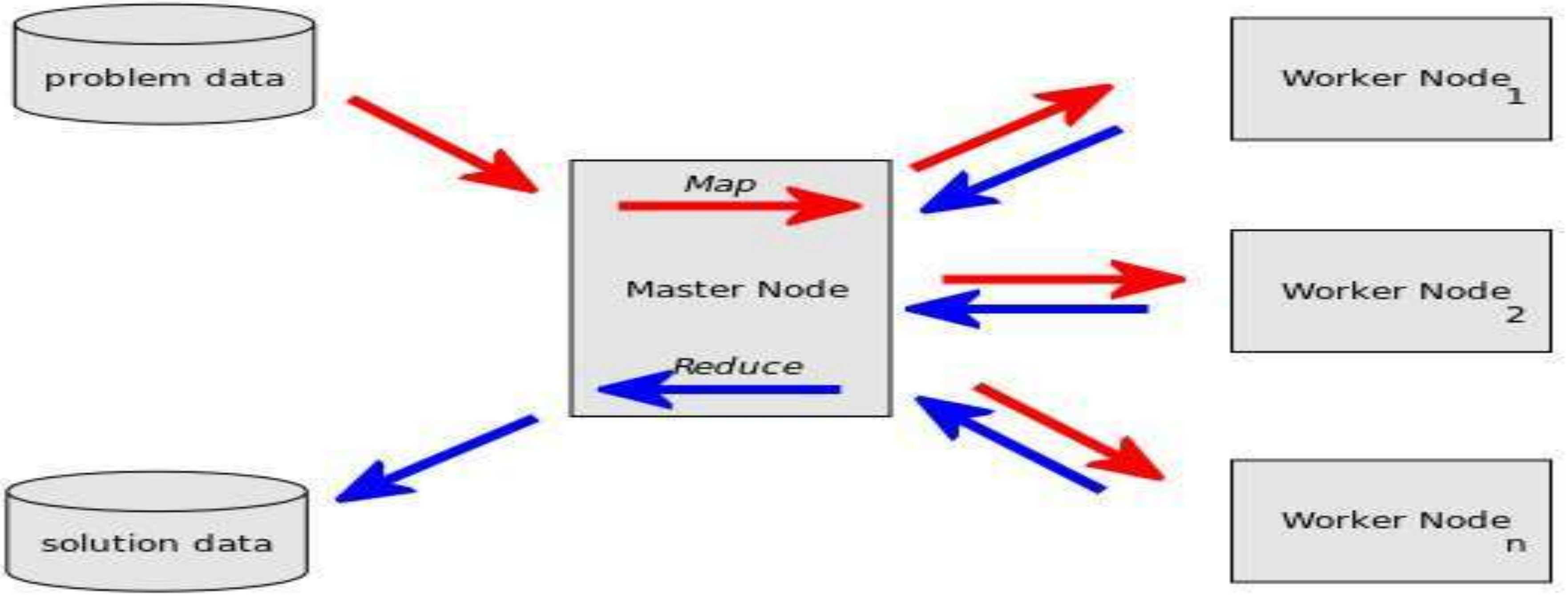
Besides, a file in an HDFS namespace is split into several blocks.
Data nodes stores these blocks. And, the name node maps the blocks to the data nodes, which handle the reading and writing operations with the file system. Furthermore, they perform tasks such as block creation, deletion etc. as instructed by the name node.

What is MapReduce ?

MapReduce is a software framework that allows writing applications to process big data simultaneously on large clusters of commodity hardware.

This framework consists of a single master job tracker and one slave task tracker per cluster node.

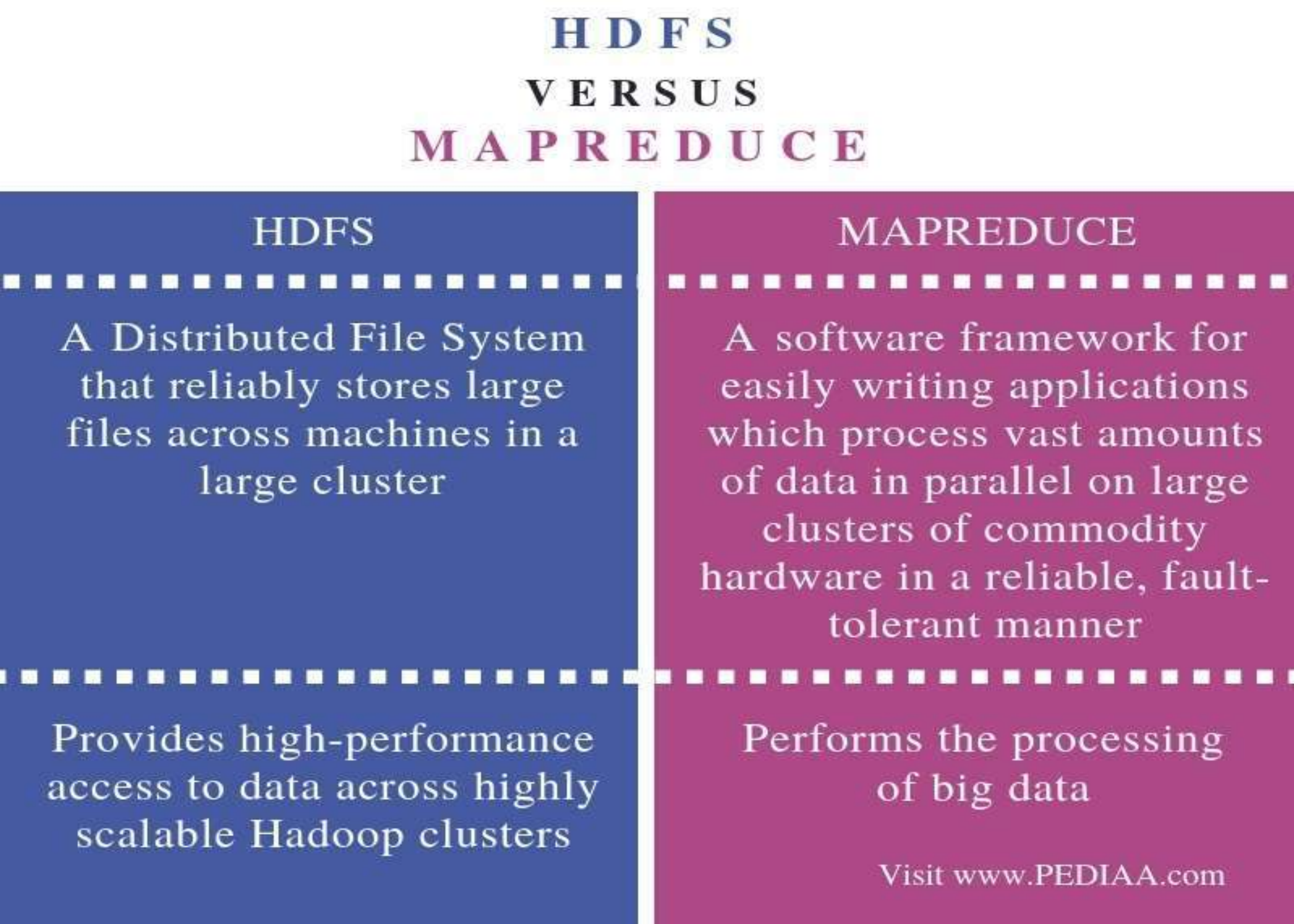
The master performs resource management, scheduling jobs on slaves, monitoring and re-executing the failed tasks. On the other hand, the slave task tracker executes the tasks instructed by the master and sends the tasks status information back to the master constantly.



Also, there are two tasks associated with MapReduce. They are the **map task** and the **reduce task**.

➢**The map task takes input data and divides them into tuples of key, value pairs while the Reduce task takes the output from a map task as input and connects those data tuples into smaller tuples.**

➢**Furthermore, the map task is performed before the reduce task.**



HDFS
VERSUS
MAPREDUCE

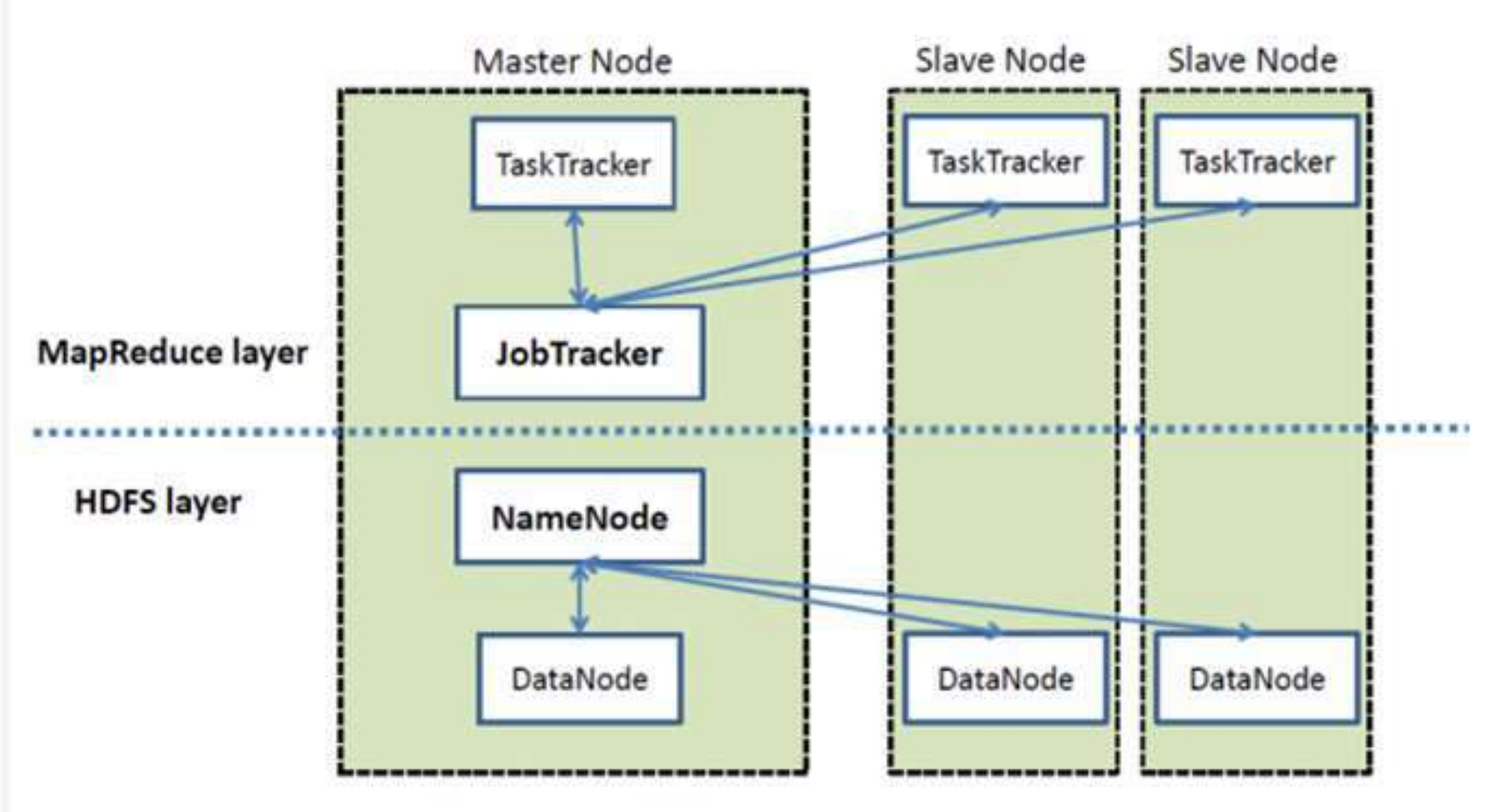
HDFS	MAPREDUCE
A Distributed File System that reliably stores large files across machines in a large cluster	A software framework for easily writing applications which process vast amounts of data in parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner
Provides high-performance access to data across highly scalable Hadoop clusters	Performs the processing of big data

Visit www.PEDIAA.com

Conclusion

In brief, HDFS and MapReduce are two modules in Hadoop architecture. The main difference between HDFS and MapReduce is that **HDFS is a distributed file system that provides high throughput access to application data** while **MapReduce is a software framework that processes big data on large clusters reliably.**

High Level Architecture of Hadoop



Hadoop distributed file system (HDFS)

- The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance, typically has a single namenode, and a cluster of datanodes forms the HDFS cluster.
- The situation is typical because each node does not require a datanode to be present. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS.
- The file system uses the TCP/IP layer for communication. Clients use Remote procedure call (RPC) to communicate between each other.

HDFS(Data Storage)

- HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines. It achieves the reliability by replicating the data across multiple hosts, and hence does not require RAID storage on hosts.
- With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack.
- Data nodes can talk to each other to rebalance data, to move copies around, and to keep the replication of data high.
- HDFS is not fully POSIX-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The tradeoff of not having a fully POSIX-compliant file-system is increased performance for data throughput and

HDFS

support for operations such as Append.

- HDFS added the high-availability capabilities for release 2.x, allowing the main metadata server (the NameNode) to be failed over manually to a backup in the event of failure, automatic fail-over.
- The HDFS file system includes secondary namenode, which regularly connects with the primary namenode and builds snapshots of the primary namenode's directory information, which the system then saves to local or remote directories.
- These checkpointed images can be used to restart a failed primary namenode without having to replay the entire journal

HDFS

of file-system actions, then to edit the log to create an up-to-date directory structure.

- Because the namenode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files.
- HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple name-spaces served by separate namenodes.

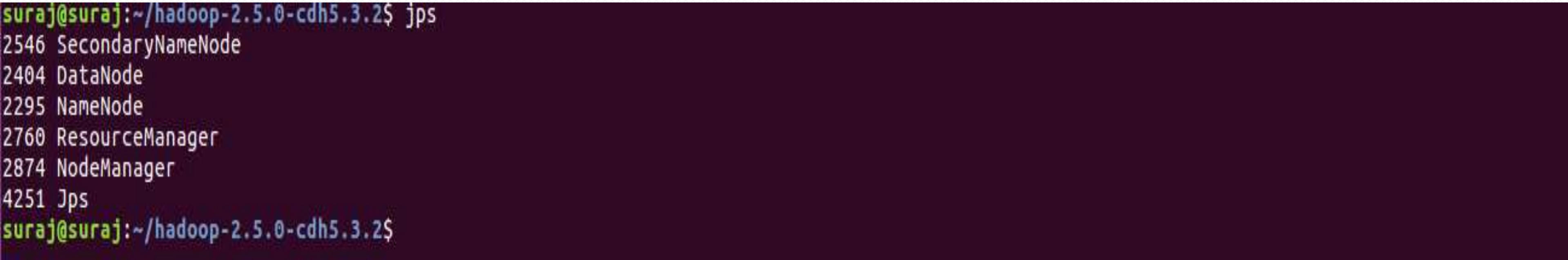
HDFS Commands

HDFS is the primary or major component of the Hadoop ecosystem which 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. To use the HDFS commands, first you need to start the Hadoop services using the following command:

```
sbin/start-all.sh
```

To check the Hadoop services are up and running use the following command:

```
jps
```



```
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$ jps
2546 SecondaryNameNode
2404 DataNode
2295 NameNode
2760 ResourceManager
2874 NodeManager
4251 Jps
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$
```

HDFS Commands

1.**ls**: This command is used to list all the files.

2.**mkdir**: To create a directory.

3.**touchz**: It creates an empty file.

4.**copyFromLocal (or) put**: To copy files/folders from local file system to hdfs store.

5.**cat**: To print file contents.

6.**copyToLocal (or) get**: To copy files/folders from hdfs store to local file system.

7.**moveFromLocal**: This command will move file from local to hdfs.

HDFS Commands

8.cp: This command is used to copy files within hdfs.

9.mv: This command is used to move files within hdfs.

10.rmr: This command deletes a file from HDFS *recursively*.

11.du: It will give the size of each file in directory.

12.dus:: This command will give the total size of directory/file.

13.stat: It will give the last modified time of directory or path. In short it will give stats of the directory or file.

14.setrep: This command is used to change the replication factor of a file/directory in HDFS. By default it is 3 for anything which is stored in HDFS (as set in hdfs *core-site.xml*).

Advantage of HDFS

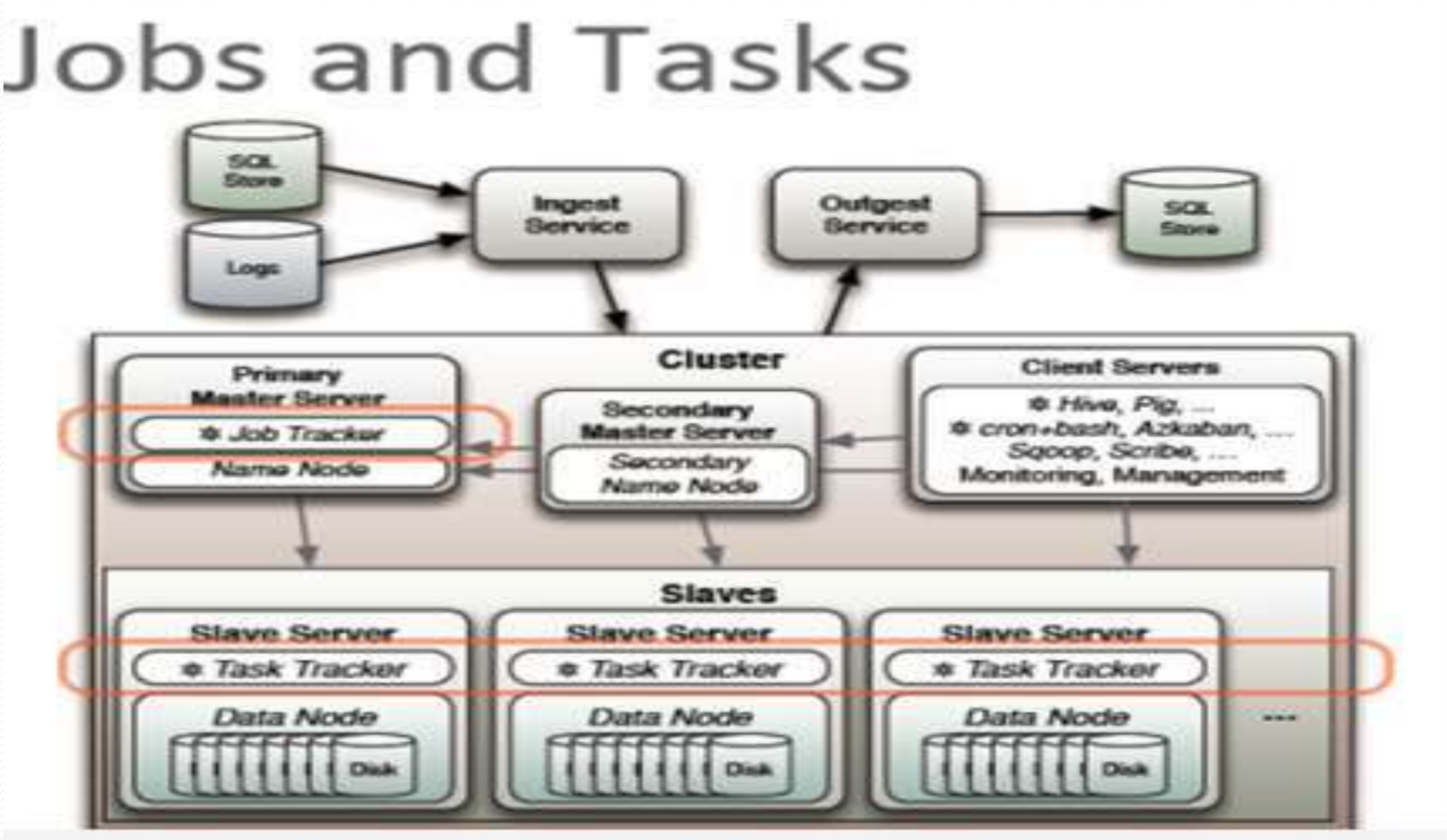
- An advantage of using HDFS is data awareness between the job tracker and task tracker.

- The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location. For example, if node A contains data (x, y, z) and node B contains data (a, b, c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z).
- This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer. When Hadoop is used with other file systems, this advantage is not always available. This can have a significant impact on job-completion times.

Limitation of HDFS

- HDFS was designed for mostly immutable files and may not be suitable for systems requiring concurrent write-operations.
- It cannot be mounted directly by an existing operating system.
- Getting data into and out of the HDFS file system, an action that often needs to be performed before and after executing a job, can be inconvenient.

JobTracker and TaskTracker: The MapReduce engine



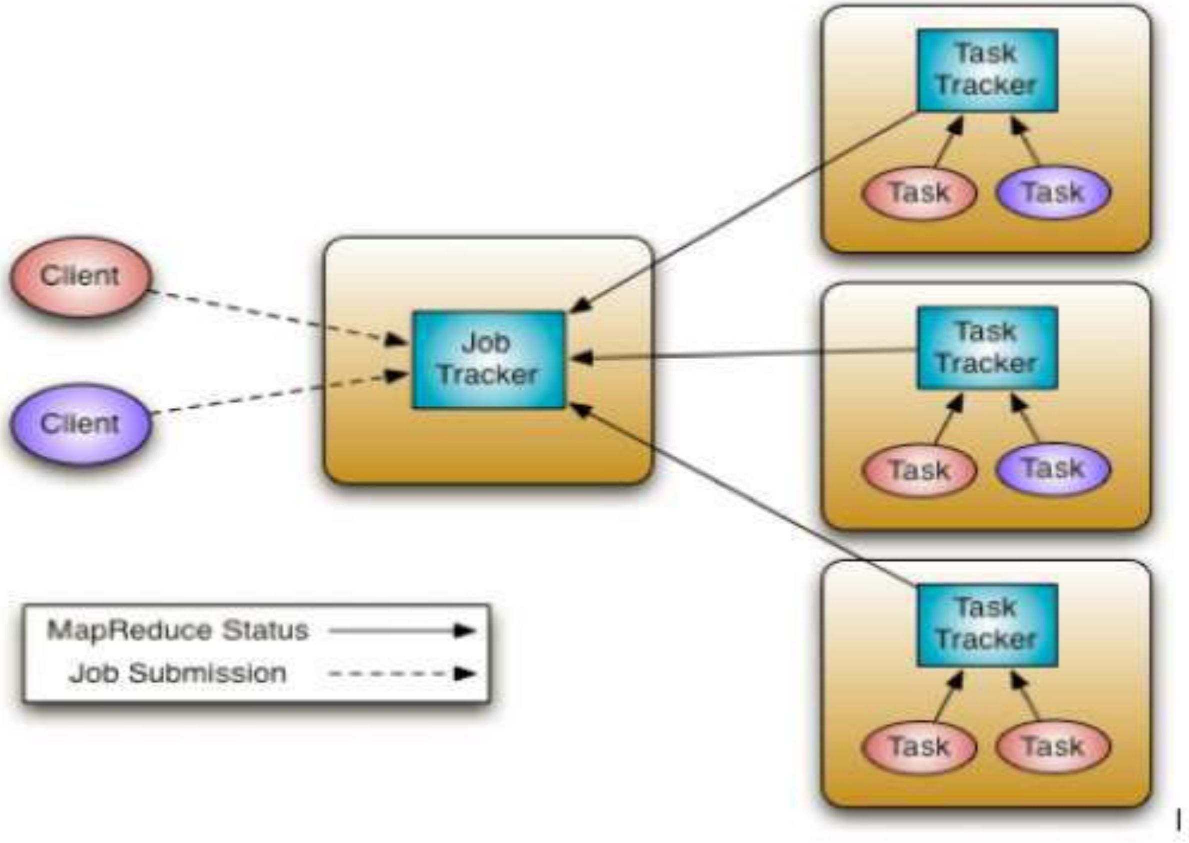
JobTracker and TaskTracker: The MapReduce engine

- The MapReduce engine, which consists of one JobTracker, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible.
- With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network.

JobTracker and TaskTracker: The MapReduce engine

- If a TaskTracker fails or times out, that part of the job is rescheduled.
- The TaskTracker on each node spawns off a separate Java Virtual Machine process to prevent the TaskTracker itself from failing if the running job crashes the JVM.
- A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status.
- The Job Tracker and TaskTracker status and information is exposed by Jetty and can be viewed from a web browser.

JobTracker and TaskTracker



JobTracker and TaskTracker: The MapReduce engine

- If the JobTracker failed on Hadoop 0.20 or earlier, all ongoing work was lost.
- Hadoop version 0.21 added some checkpointing to this process.
- The JobTracker records what it is up to in the file system.
- When a JobTracker starts up, it looks for any such data, so that it can restart work from where it left off.

Limitations of this approach in Hadoop 1.x

- The allocation of work to TaskTrackers is very simple. Every TaskTracker has a number of available slots (such as "4 slots"). Every active map or reduce task takes up one slot.

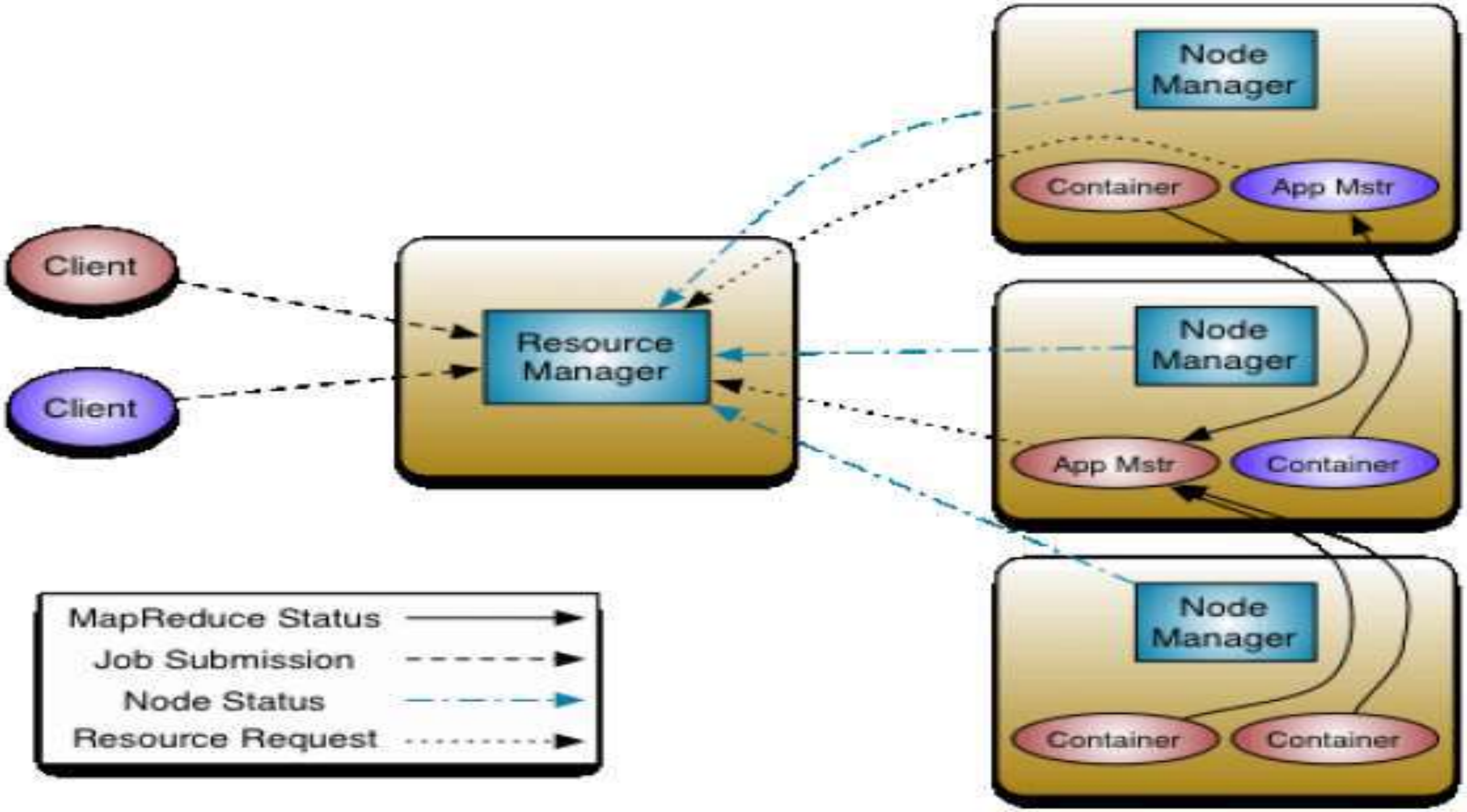
- The Job Tracker allocates work to the tracker nearest to the data with an available slot. There is no consideration of the current system load of the allocated machine, and hence its actual availability.

- If one TaskTracker is very slow, it can delay the entire MapReduce job—especially towards the end of a job, where everything can end up waiting for the slowest task. With speculative execution enabled, however, a single task can be executed on multiple slave nodes.

Apache Hadoop NextGen MapReduce (YARN)

- YARN is a sub-project of Hadoop at the Apache Software Foundation introduced in Hadoop 2.0 that separates the resource management and processing components.
- YARN was born of a need to enable a broader array of interaction patterns for data stored in HDFS beyond MapReduce.
- The YARN-based architecture of Hadoop 2.0 provides a more general processing platform that is not constrained to MapReduce.

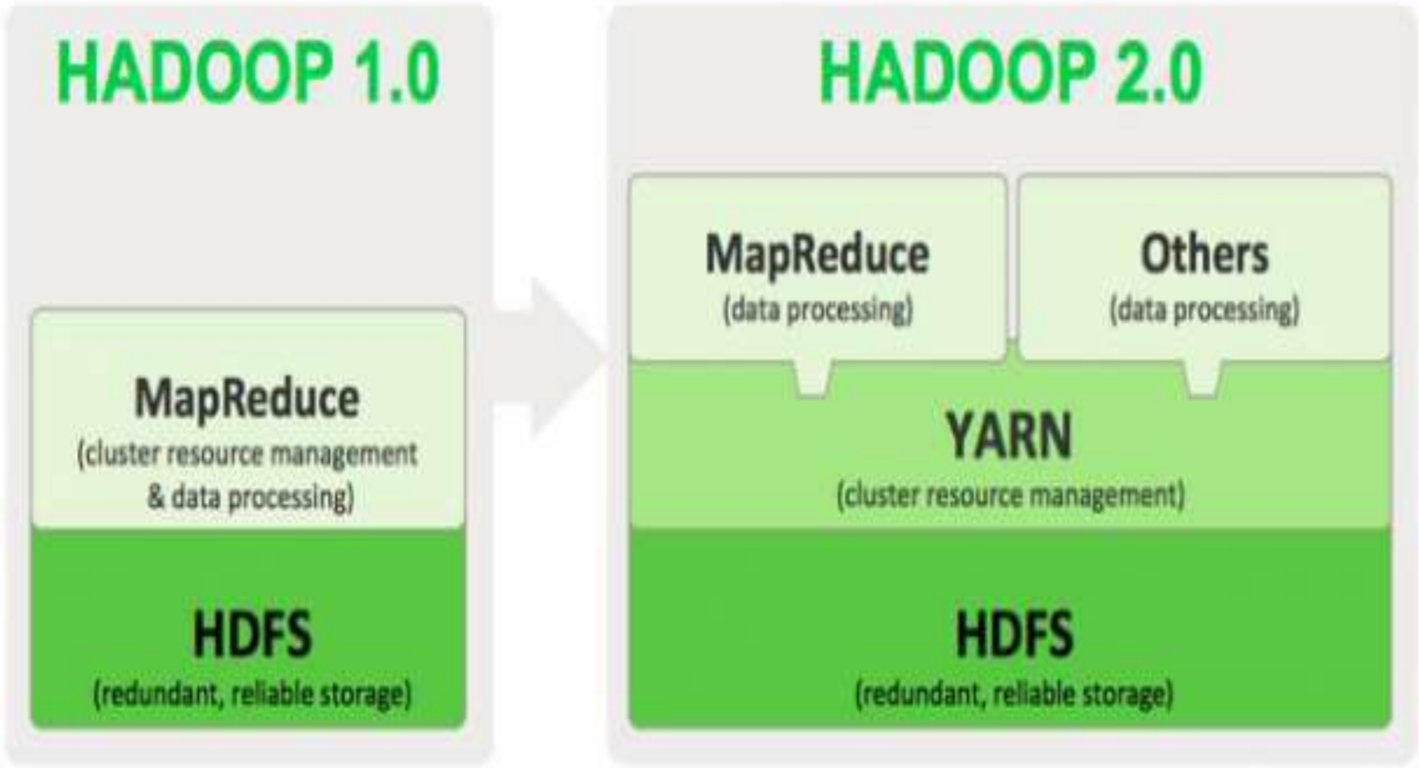
Resource Manager



Apache Hadoop NextGen MapReduce (YARN)

- The fundamental idea of MRv2 is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons.
- The idea is to have a global Resource Manager (RM) and per-application Application Master (AM). An application is either a single job in the classical sense of Map-Reduce jobs or a DAG of jobs.
- The Resource Manager and per-node slave, the Node Manager (NM), form the data-computation framework. The Resource Manager is the ultimate authority that arbitrates resources among all the applications in the system.

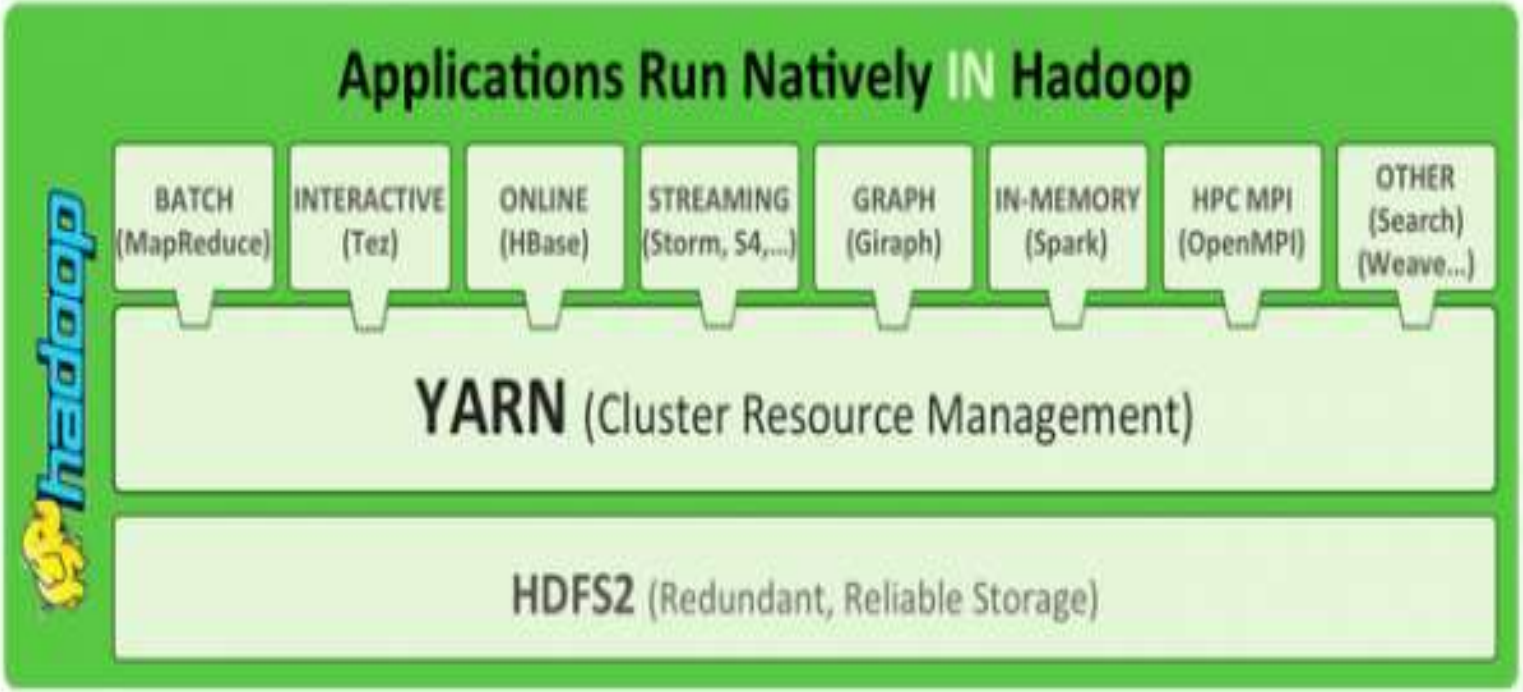
Hadoop 1.0 and 2.0


HADOOP 1.0
MapReduce
(cluster resource management & data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)

YARN

- As part of Hadoop 2.0, YARN takes the resource management capabilities that were in MapReduce and packages them so they can be used by new engines.This also streamlines MapReduce to do what it does best, process data.
- With YARN, you can now run multiple applications in Hadoop, all sharing a common resource management. Many organizations are already building applications on YARN in order to bring them in to Hadoop.
- When enterprise data is made available in HDFS, it is important to have multiple ways to process that data. With Hadoop 2.0 and YARN organizations can use Hadoop for streaming, interactive and a world of other Hadoop based applications.

Applications run in Hadoop



Applications Run Natively IN Hadoop
hadoop
BATCH (MapReduce)
INTERACTIVE (Tez)
ONLINE (HBase)
STREAMING (Storm, S4,...)
GRAPH (Giraph)
IN-MEMORY (Spark)
HPC MPI (OpenMPI)
OTHER (Search) (Weave...)
YARN (Cluster Resource Management)
HDFS2 (Redundant, Reliable Storage)

How YARN works

The fundamental idea of YARN is to split up the two major responsibilities of the JobTracker/TaskTracker into separate entities:

- a global Resource Manager
- a per-application Application Master
- a per-node slave Node Manager and
- a per-application container running on a Node Manager

Cont..

- The Resource Manager and the Node Manager form the new, and generic, system for managing applications in a distributed manner.
- The Resource Manager is the ultimate authority that arbitrates resources among all the applications in the system.
- The per-application Application Master is a framework-specific entity and is tasked with negotiating resources from the Resource Manager and working with the Node Manager(s) to execute and monitor the component tasks.
- The Resource Manager has a scheduler, which is responsible for allocating resources to the various running applications, according to constraints such as queue capacities, user-limits etc.

Cont..

- The scheduler performs its scheduling function based on the resource requirements of the applications.
- The Node Manager is the per-machine slave, which is responsible for launching the applications' containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the Resource Manager.
- Each Application Master has the responsibility of negotiating appropriate resource containers from the scheduler, tracking their status, and monitoring their progress.
- From the system perspective, the Application Master runs as a normal container.

Difference between HADOOP 1 and HADOOP 2

HADOOP is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on Java programming with some native code in C and shell scripts.

1. Components: In Hadoop 1 we have MapReduce but Hadoop 2 has YARN(Yet Another Resource Negotiator) and MapReduce version 2.

HADOOP 1	HADOOP 2
HDFS	HDFS
Map Reduce	YARN / MRv2

Cont…

2. Daemons:

HADOOP 1	HADOOP 2
Namenode	Namenode
Datanode	Datanode
Secondary Namenode	Secondary Namenode
Job Tracker	Resource Manager
Task Tracker	Node Manager

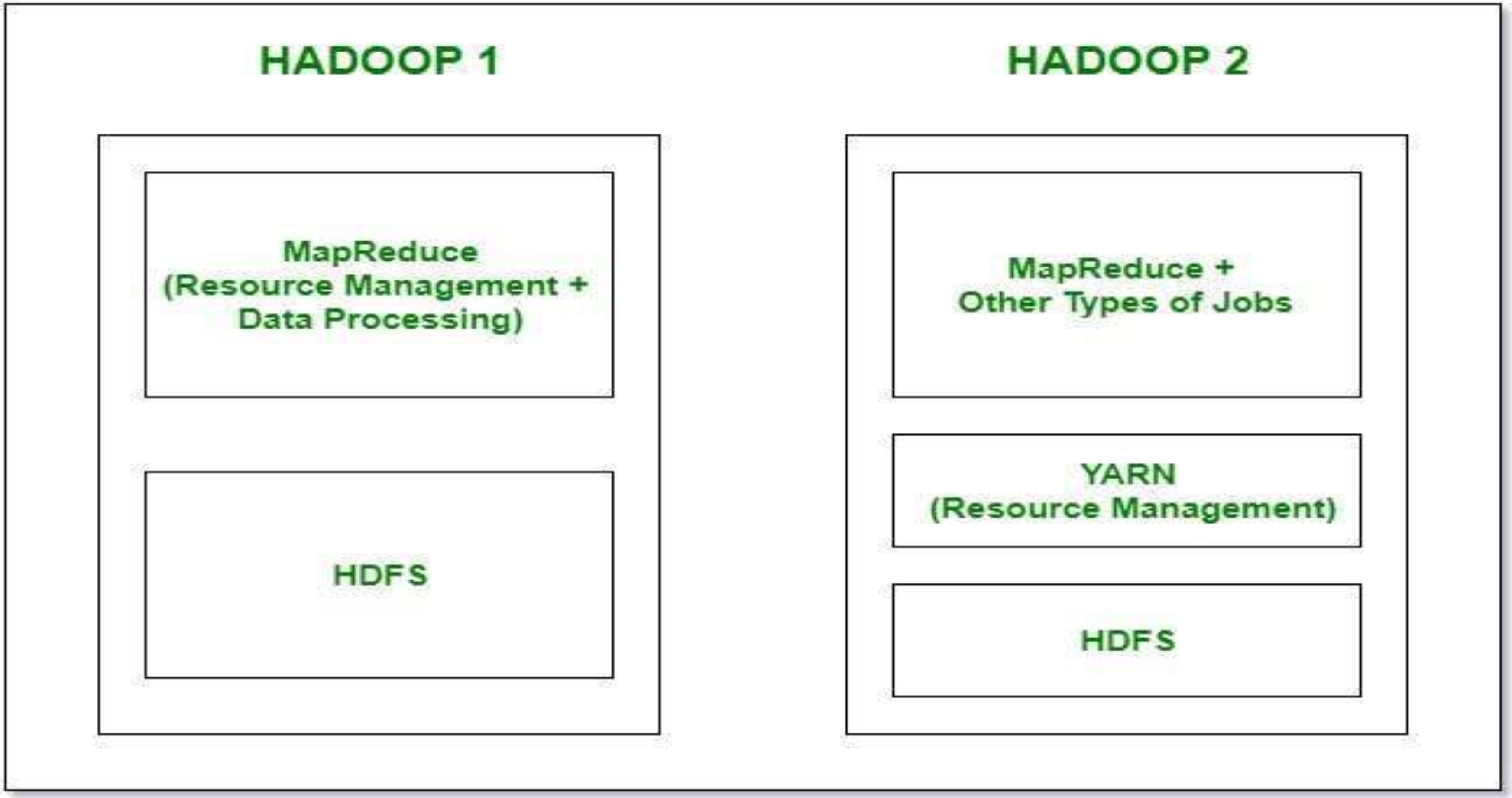
Cont…

3. Working:

- In *Hadoop 1*, there is HDFS which is used for storage and top of it, Map Reduce which works as Resource Management as well as Data Processing. Due to this workload on Map Reduce, it will affect the performance.

- In *Hadoop 2*, there is again HDFS which is again used for storage and on the top of HDFS, there is YARN which works as Resource Management. It basically allocates the resources and keeps all the things going on.

3. Working:


HADOOP 1
MapReduce
(Resource Management +
Data Processing)
HDFS
HADOOP 2
MapReduce +
Other Types of Jobs
YARN
(Resource Management)
HDFS

Cont…

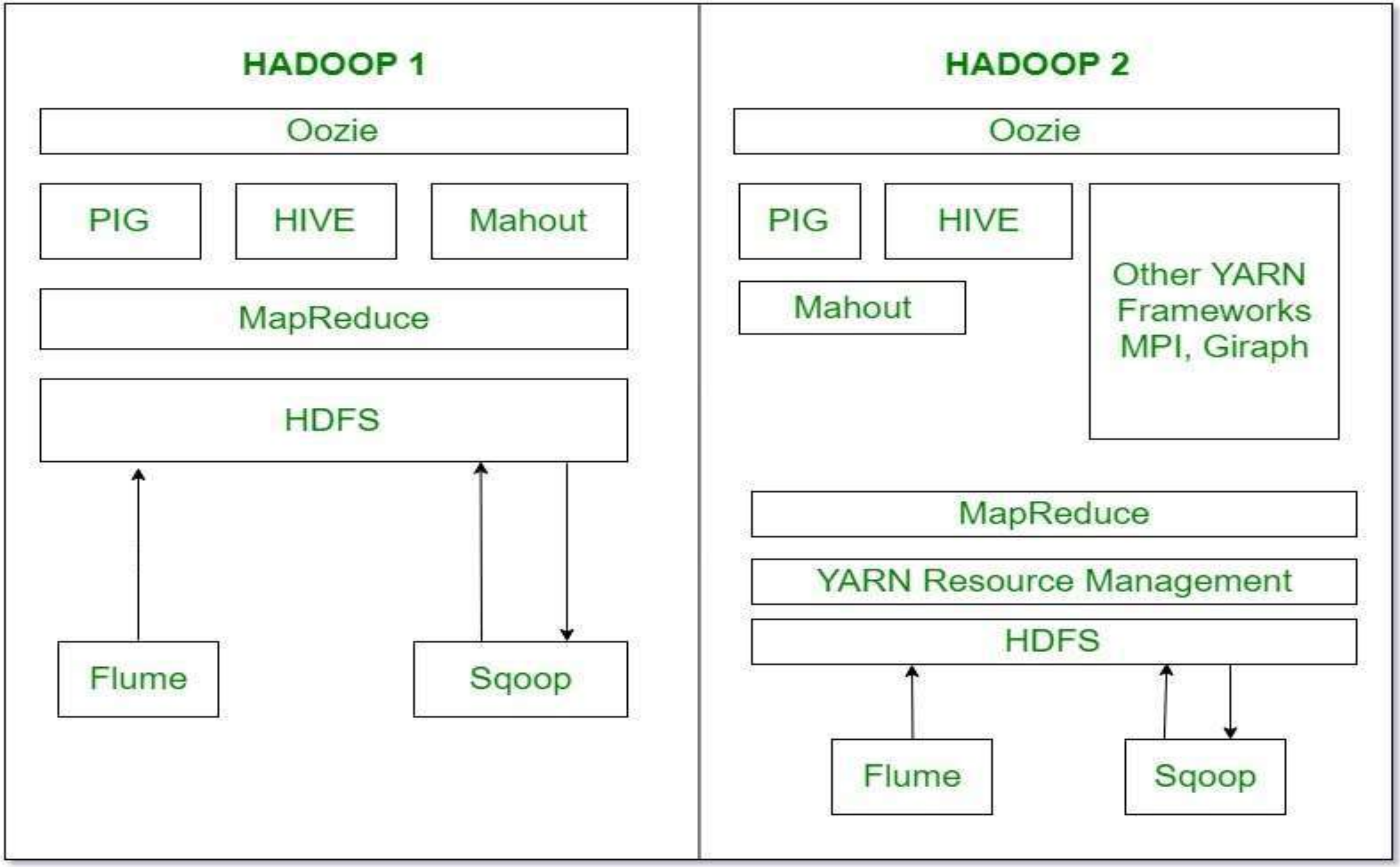
4. Limitations:

- *Hadoop 1* is a Master-Slave architecture. It consists of a single master and multiple slaves. Suppose if master node got crashed then irrespective of your best slave nodes, your cluster will be destroyed. Again for creating that cluster means copying system files, image files, etc. on another system is too much time consuming which will not be tolerated by organizations in today's time.

- *Hadoop 2* is also a Master-Slave architecture. But this consists of multiple masters (i.e active namenodes and standby namenodes) and multiple slaves. If here master node got crashed then standby master node will take over it. You can make multiple combinations of active-standby nodes. *Thus Hadoop 2 will eliminate the problem of a single point of failure*.

Cont…

5. Ecosystem



5. Ecosystem(Tools)

➢Oozie is basically Work Flow Scheduler. It decides the particular time of jobs to execute according to their dependency.

➢Pig, Hive and Mahout are data processing tools that are working on the top of Hadoop.

➢Sqoop is used to import and export structured data. You can directly import and export the data into HDFS using SQL database.

➢Flume is used to import and export the unstructured data and streaming data.

References:-

1.https://www.techopedia.com/

2. https://pediaa.com/

3.https://www.geeksforgeeks.org/

THANKYOU