UNIT-4
Understanding Hadoop Ecosystem

Prof. Vipul Gamit

PIET

Parul University

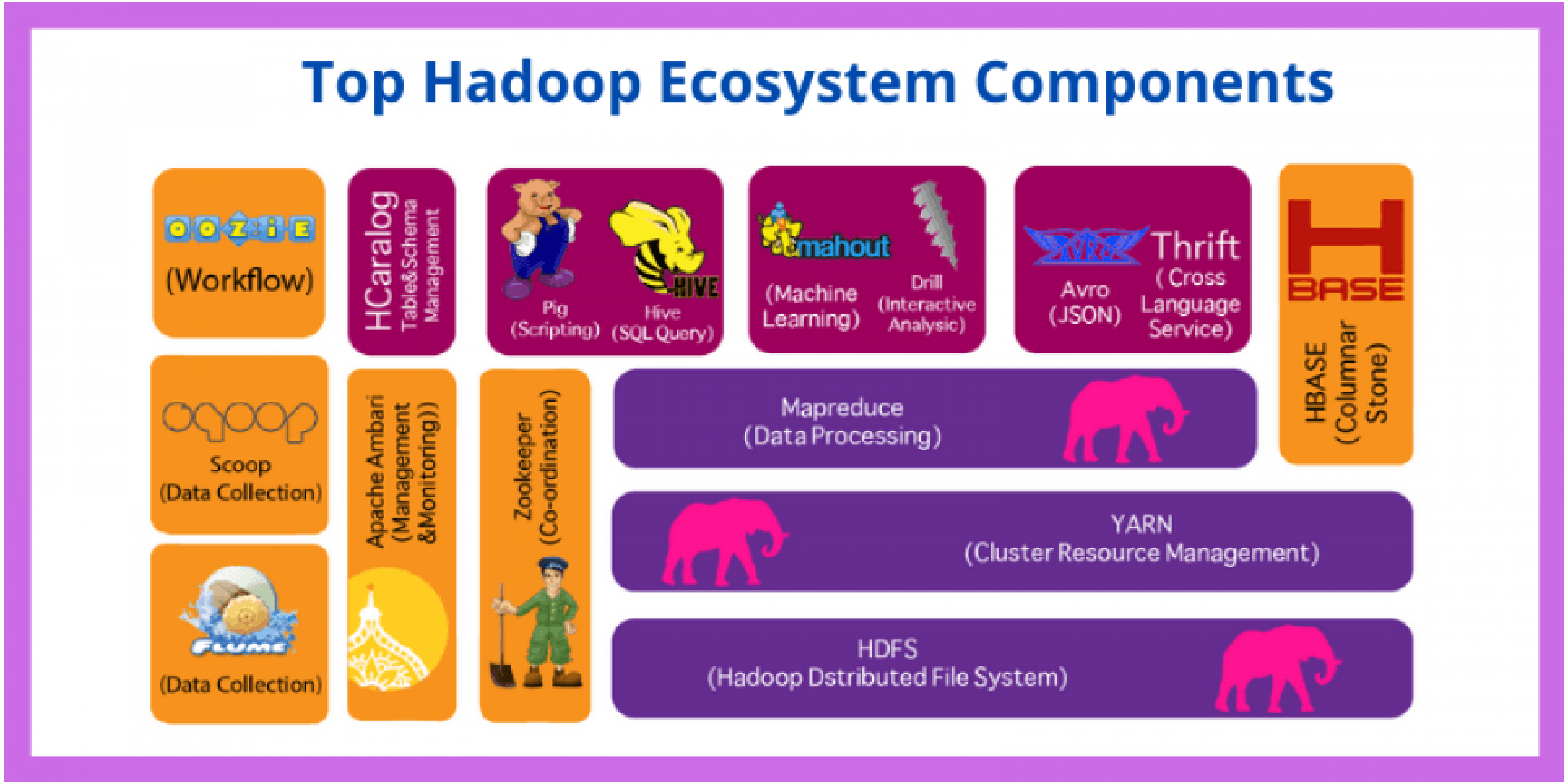
History of Hadoop

- Apache top level project, open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.
- It is a flexible and highly-available architecture for large scale computation and data processing on a network of commodity hardware.
- Hadoop was developed to support distribution for the search engine project.
- The project was funded by Yahoo then it gave the project to Apache Software Foundation.

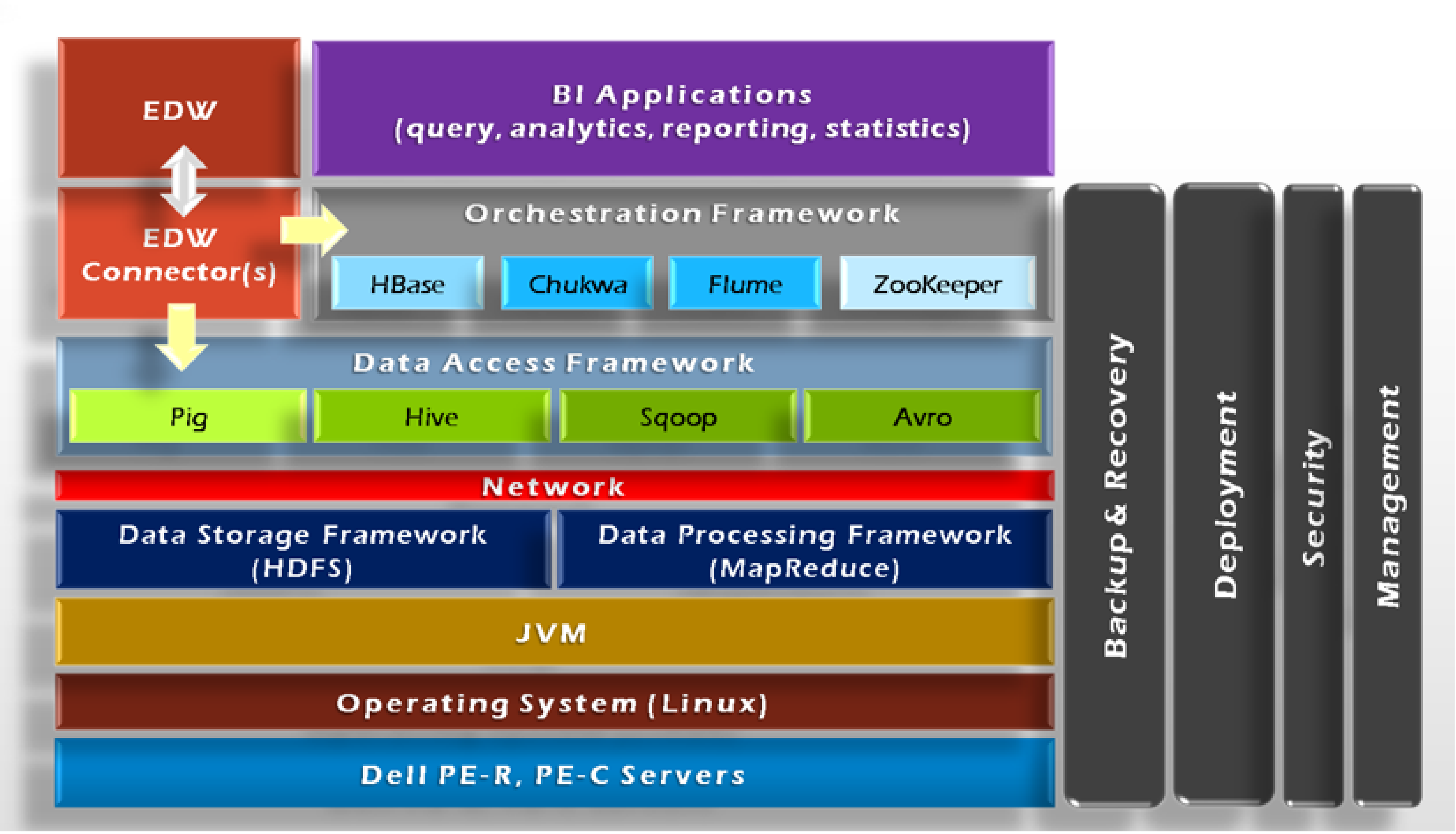
Hadoop Ecosystem

- The Hadoop Ecosystem is a framework that enables distributed storage and processing of huge amounts of data in a cost-effective way.
- Hadoop refers to an ecosystem of related products.
- The two Hadoop core components are HDFS and MapReduce.
- The Oracle BDA uses the Hadoop Ecosystem to:
  - Store and process data
  - Provide scaling without limits
  - Solve the various problems that are encountered in big data

Hadoop Ecosystem



Hadoop Ecosystem



Hadoop Ecosystem

- Apache Hadoop contains two main core components:
  - Hadoop Distributed File System (HDFS) is a distributed file system for storing information and it sits on top of the OS that you are using.
  - MapReduce is a parallel processing framework that operates on local data whenever possible. It abstracts the complexity of parallel processing. This enables developers to focus more on the business logic rather than on the processing framework.
  - Hadoop enables parallel processing of large data sets because the distributed file system has been designed to work in conjunction with the distributed processing framework. This allows for clusters with thousands of nodes.
  - HDFS stores large files. It breaks those files into blocks of 64, 128, or 256 megabytes (MB)

Hadoop Ecosystem

- These blocks are then distributed across the cluster and replicated numerous times. The default replication factor is 3. This replication has two main purposes:
  1. The redundancy yields fault tolerance. If a server fails, the other two servers automatically take over its workload.

  2. It allows for collocating processing (MapReduce) and data (HDFS). The goal is to move the processing to the data to achieve better performance.
- Hadoop is an open source software developed by the Apache Software Foundation (ASF).
- Cloudera is a company that provides support, consulting, and management tools for Hadoop.
- Cloudera also has a distribution of software called Cloudera’s Distribution Including Apache Hadoop (CDH).

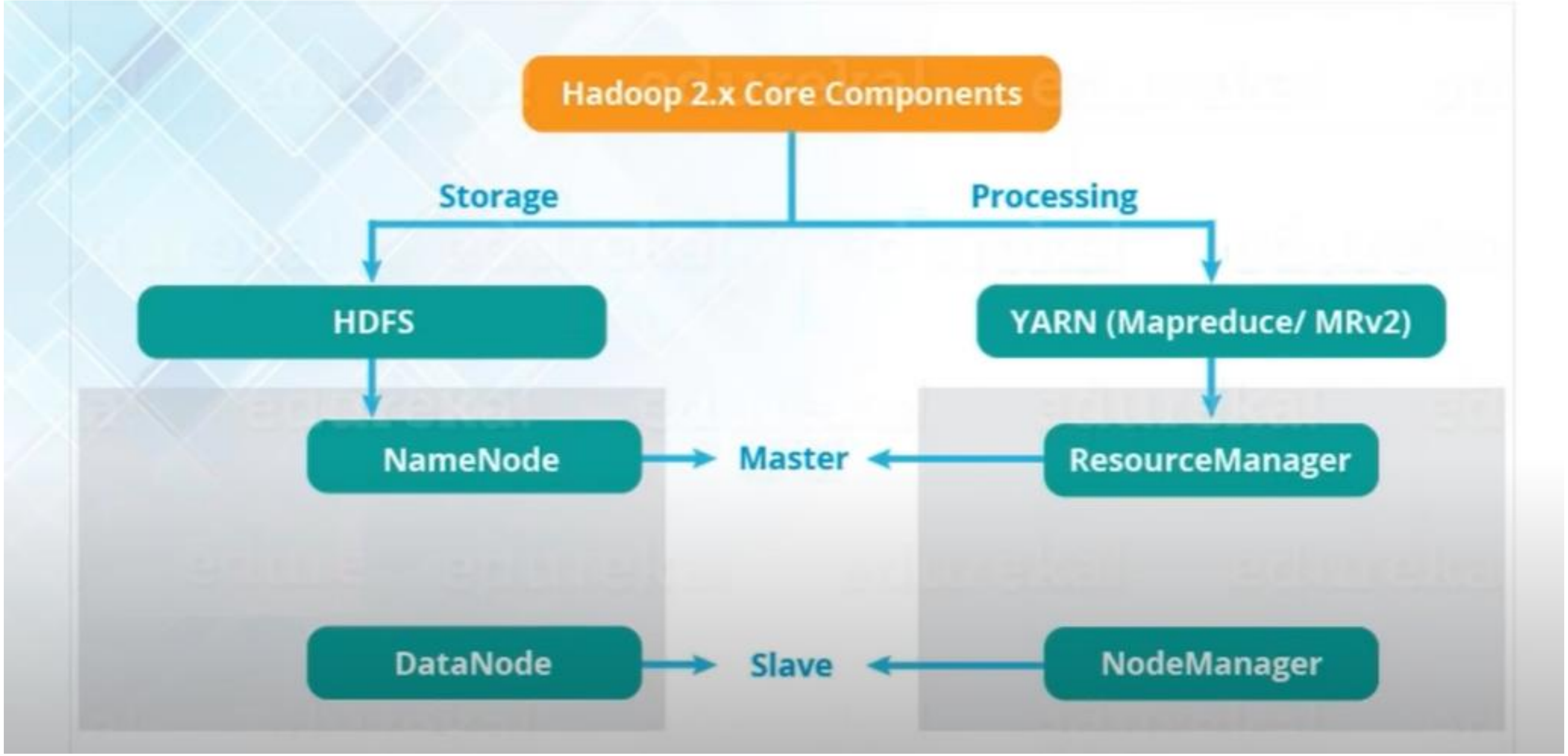
HADOOP ECOSYSTEM

- Hadoop Ecosystem is neither a programming language nor a service, it is a platform or framework which solves big data problems. You can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.
- Below are the Hadoop components, that together form a Hadoop ecosystem.
  - HDFS -> Hadoop Distributed File System
  - YARN -> Yet Another Resource Negotiator
  - MapReduce -> Data processing using programming
  - Spark -> In-memory Data Processing
  - PIG, HIVE-> Data Processing Services using Query (SQL-like)
  - HBase -> NoSQL Database
  - Mahout, Spark MLlib -> Machine Learning
  - Apache Drill -> SQL on Hadoop
  - Zookeeper -> Managing Cluster
  - Oozie -> Job Scheduling
  - Flume, Sqoop -> Data Ingesting Services
  - Solr& Lucene -> Searching & Indexing
  - Ambari -> Provision, Monitor and Maintain cluster

HDFS

- Hadoop Distributed File System is the core component or you can say, the backbone of Hadoop Ecosystem.
- HDFS is the one, which makes it possible to store different types of large data sets (i.e. structured, unstructured and semi structured data).
- HDFS creates a level of abstraction over the resources, from where we can see the whole HDFS as a single unit.
- It helps us in storing our data across various nodes and maintaining the log file about the stored data (metadata).
- HDFS has two core components, i.e. NameNode and DataNode.
    1. The NameNode is the main node and it doesn’t store the actual data. It contains metadata, just like a log file or you can say as a table of content. Therefore, it requires less storage and high computational resources.
    2. On the other hand, all your data is stored on the DataNodes and hence it requires more storage resources. These DataNodes are commodity hardware (like your laptops and desktops) in the distributed environment. That’s the reason, why Hadoop solutions are very cost effective.
    3. You always communicate to the NameNode while writing the data. Then, it internally sends a request to the client to store and replicate data on various DataNodes.

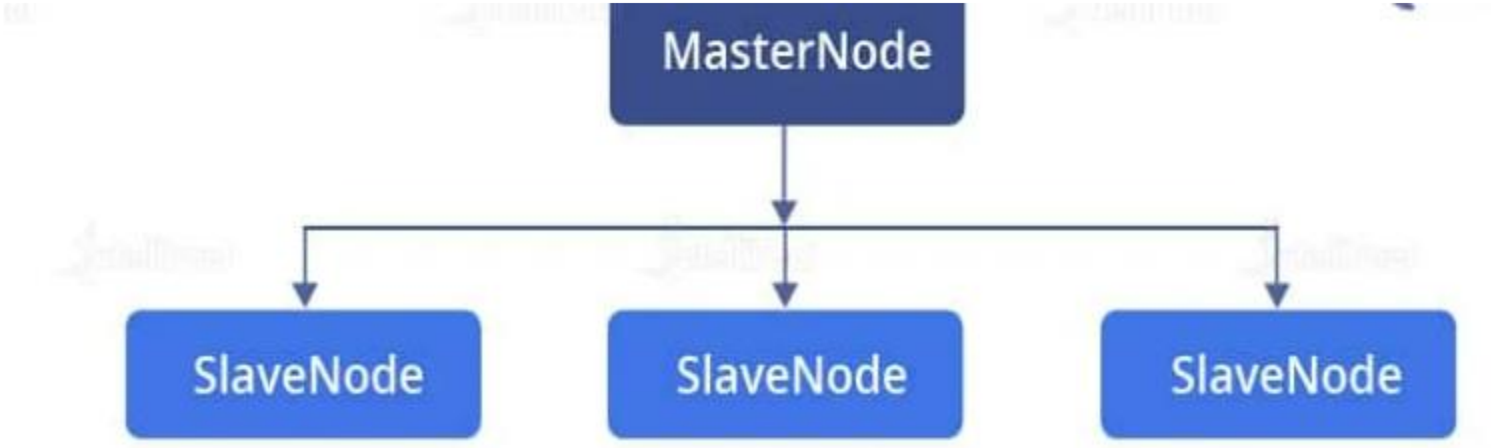
HDFS



Hadoop 2.x Core Components
Storage
Processing
HDFS
YARN (Mapreduce/ MRv2)
NameNode
Master
ResourceManager
DataNode
Slave
NodeManager

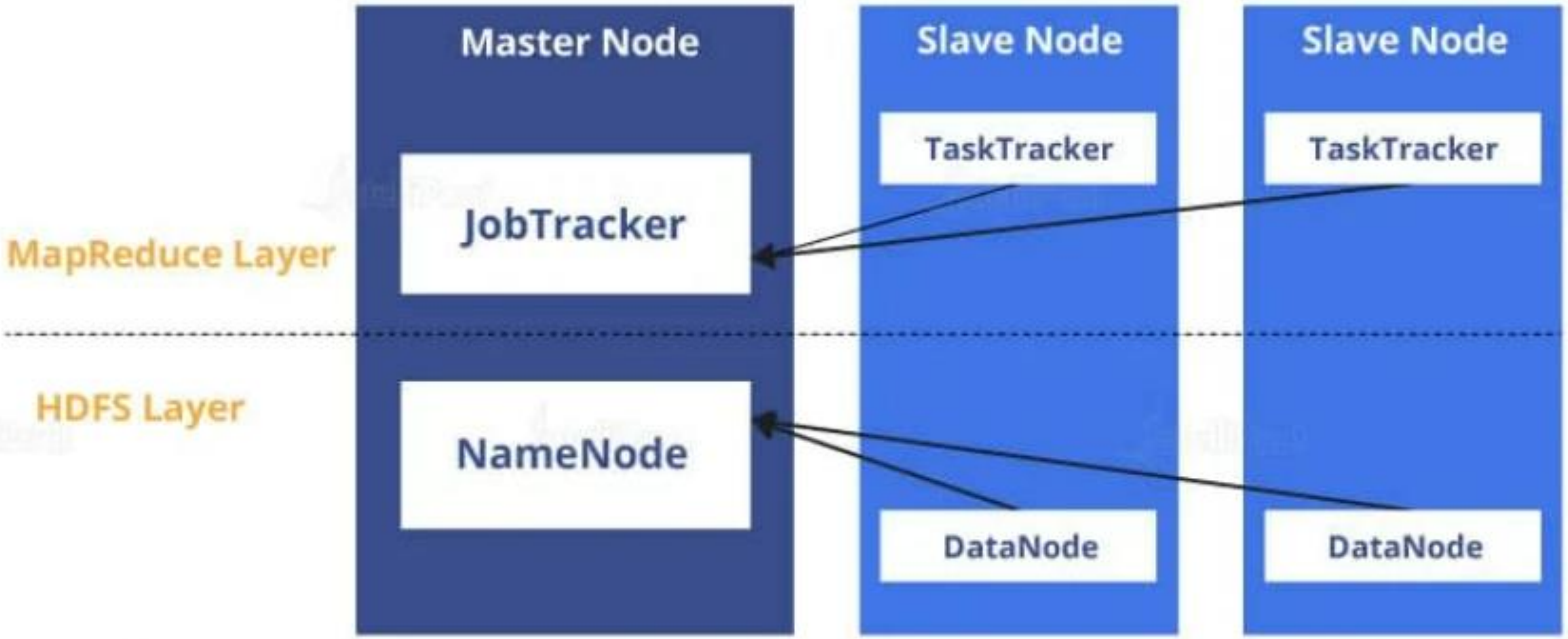
Hadoop Architecture

- Hadoop follows a master-slave architecture for storing data and data processing. This master-slave architecture has master nodes and slave nodes as shown in the image below:



Hadoop Architecture

- First, we will see how data is stored in Hadoop, and then we will move on to how it is processed. While talking about the storage of files in Hadoop, HDFS comes into place.



Hadoop Architecture

Understanding the architecture:

- **NameNode**: NameNode is basically a master node that acts like a monitor and supervises operations performed by DataNodes.
- **Secondary NameNode**: A Secondary NameNode plays a vital role in case if there is some technical issue in the NameNode.
- **DataNode**: DataNode is the slave node that stores all files and processes.
- **Mapper**: Mapper maps data or files in the DataNodes. It will go to every DataNode and run a particular set of codes or operations in order to get the work done.
- **Reducer**: While a Mapper runs a code, a Reducer is required for getting the result from each Mapper.
- **JobTracker**: JobTracker is a master node used for getting the location of a file in different DataNodes. It is a very important service in Hadoop as if it goes down, all the running jobs will get halted.
- **TaskTracker**: TaskTracker is a reference for the JobTracker present in the DataNodes. It accepts different tasks, such as map, reduces, and shuffle operations, from the JobTracker. It is a key player performing the main MapReduce functions.
- **Block:** Block is a small unit wherein the files are split. It has a default size of 64 MB and can be increased as needed.
- **Cluster**: A cluster is a set of machines such as DataNodes, NameNodes, Secondary NameNodes, etc.

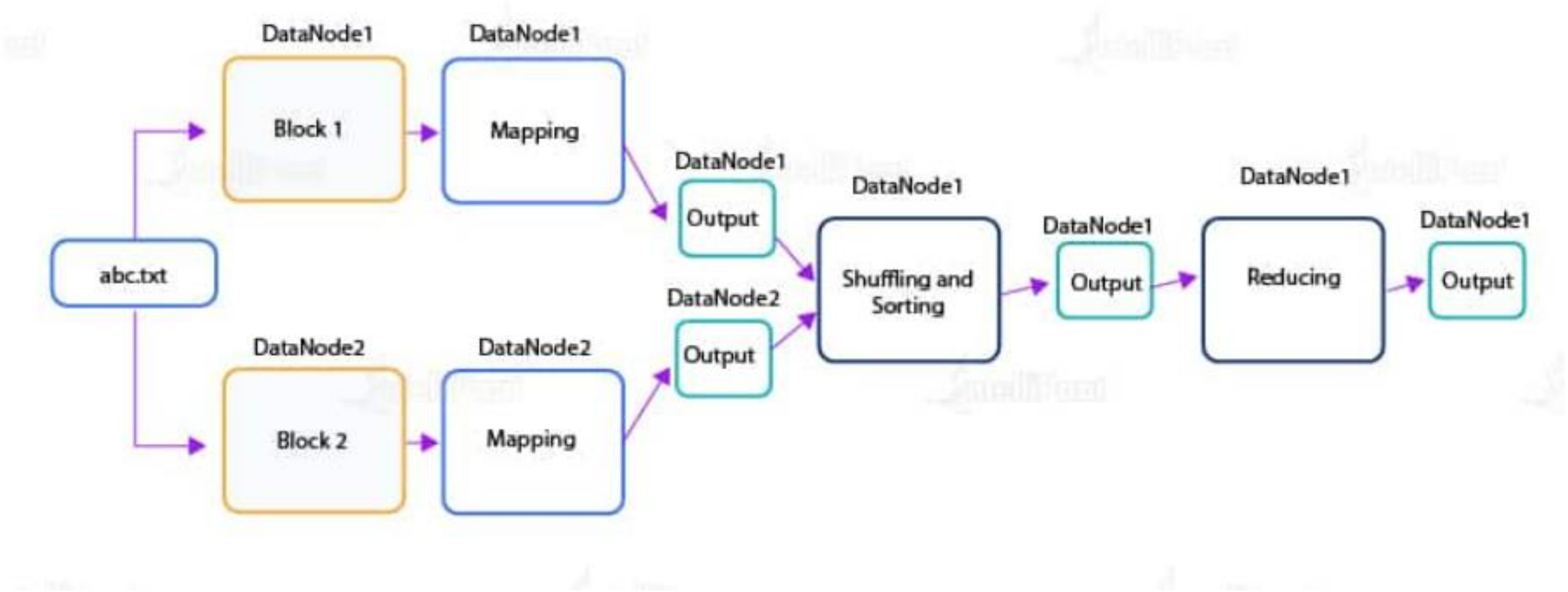
Requirements:

- Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
  - Structured and non-structured data
  - Simple programming models
- High scalability and availability
- Use commodity (cheap!) hardware with little redundancy
- Fault-tolerance
- Move computation rather than data

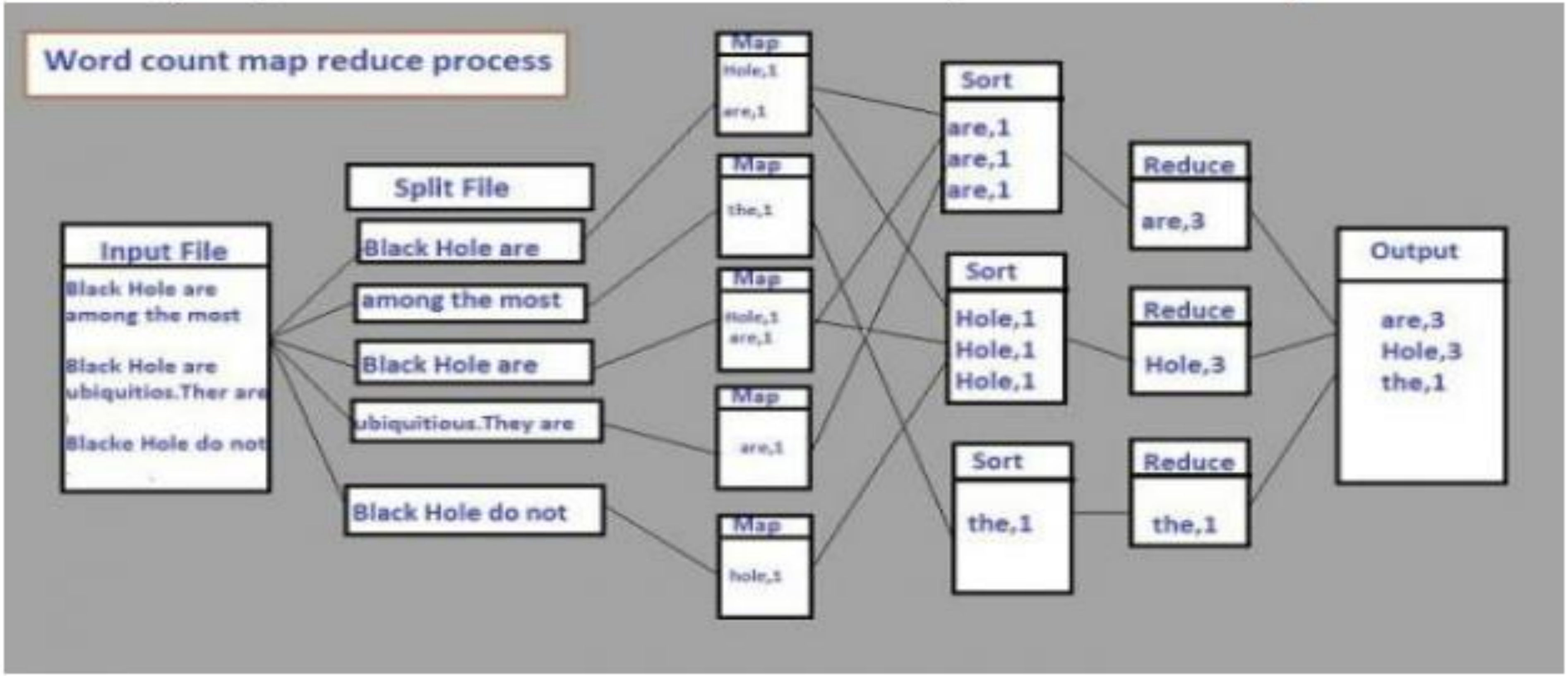
MapReduce Layer

- MapReduce is a patented software framework introduced by Google to support distributed computing on large datasets on clusters of computers.
- It is basically an operative programming model that runs in the Hadoop background providing simplicity, scalability, recovery, and speed, including easy solutions for data processing.
- This MapReduce framework is proficient in processing a tremendous amount of data parallelly on large clusters of computational nodes.
- MapReduce is a programming model that allows you to process your data across an entire cluster.
- It basically consists of Mappers and Reducers that are different scripts you write or different functions you might use when writing a MapReduce program.
- Mappers have the ability to transform your data in parallel across your computing cluster in a very efficient manner; whereas, Reducers are responsible for aggregating your data together.
- Mappers and Reducers put together can be used to solve complex problems.

MapReduce Layer



Word Count map reduce process



Hadoop MapReduce Word Count Process

Analyzing Data with Unix tools

- Through the use of Unix tools: Software developers can quickly explore and modify code, data, and tests.
- IT professionals can scrutinize log files, network traces, performance figures, filesystems and the behavior of processes.
- Data analysts can extract, transform, filter, process, load, and summarize huge data sets.

Hadoop Streaming

- Both the mapper and the reducer are python scripts that read the input from standard input and emit the output to standard output.
- The utility will create a Map/Reduce job, submit the job to an appropriate cluster, and monitor the progress of the job until it completes.
- When a script is specified for mappers, each mapper task will launch the script as a separate process when the mapper is initialized.
- As the mapper task runs, it converts its inputs into lines and feed the lines to the standard input (STDIN) of the process.
- In the meantime, the mapper collects the line-oriented outputs from the standard output (STDOUT) of the process and converts each line into a key/value pair, which is collected as the output of the mapper.
- By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) will be the value.
- If there is no tab character in the line, then the entire line is considered as the key and the value is null. However, this can be customized, as per one need.
- When a script is specified for reducers, each reducer task will launch the script as a separate process, then the reducer is initialized.
- As the reducer task runs, it converts its input key/values pairs into lines and feeds the lines to the standard input (STDIN) of the process.
- In the meantime, the reducer collects the line-oriented outputs from the standard output (STDOUT) of the process, converts each line into a key/value pair, which is collected as the output of the reducer.
- By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) is the value. However, this can be customized as per specific requirements.

IBM Big Data Strategy

- Smarter Planet was a corporate initiative of IBM, which sought to highlight how government and business leaders were capturing the potential of smarter systems to achieve economic and sustainable growth and societal progress. In November 2008, in his speech at the Council on Foreign Relations, IBM’s Chairman, CEO and President Sam Palmisano, outlined an agenda for building a ‘Smarter Planet’. He emphasized how the world’s various systems – like traffic, water management, communication technology, smart grids, healthcare solutions, and rail transportation – were struggling to function effectively