

# Big Data and Hadoop

## Lesson 1: Introduction to Big Data and Hadoop



# Objectives

By the end of this lesson, you will be able to:

- Explain the characteristics of Big Data
- Describe the basics of Hadoop and HDFS architecture
- List the features and processes of MapReduce
- Describe the basics of Pig



Following are the reasons why Big Data is needed:

- 90% of the data in the world today has been created in the last two years alone.
- 80% of the data is unstructured or exists in widely varying structures which are difficult to analyze.
- Structured formats have some limitations with respect to handling large quantities of data.
- It is difficult to integrate information distributed across multiple systems.
- Most business users do not know what should be analyzed.
- Potentially valuable data is dormant or discarded.
- It is too expensive to integrate large volumes of unstructured data.
- A lot of information has a short, useful lifespan.
- Context adds meaning to the existing information.

Big Data has three characteristics: variety, velocity, and volume.

## Variety

Variety encompasses managing the complexity of data in many different structures, ranging from relational data to logs and raw text.

## Velocity

Velocity accounts for the streaming of data and movement of large volume of data at a high speed.

## Volume

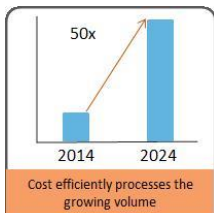
Volume denotes the scaling of data ranging from terabytes to zettabytes.

# Characteristics of Big Data Technology

Following are the characteristics of Big Data technology:

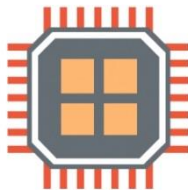
Cost efficiently processes the growing volume

- Turned 12 terabytes of Tweets created each day into improved product sentiment analysis
- Converted 350 billion annual meter readings to better predict power consumption



Responds to the increasing velocity

- Scrutinized 5 million trade events created each day to identify potential frauds
- Analyzed 500 million daily call detail records in real time to predict customer churn faster



Collectively analyzes the widening variety

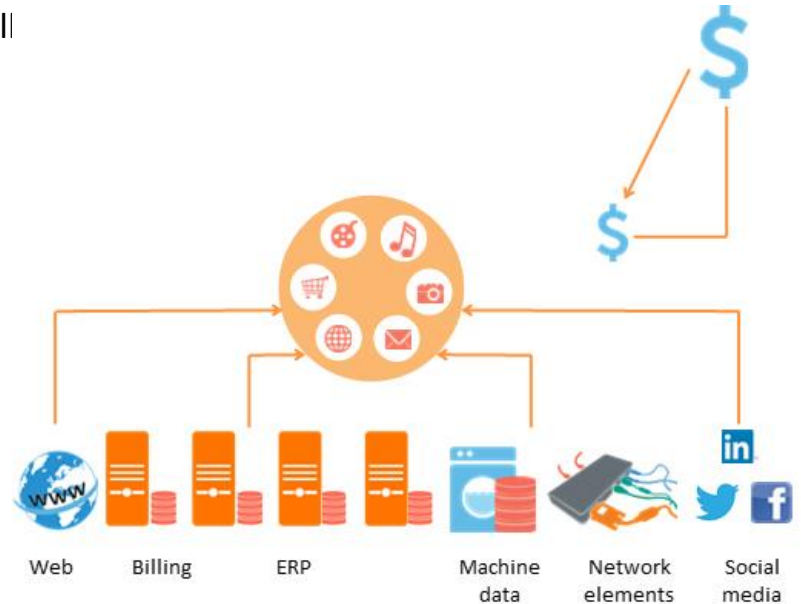
- Monitored hundreds of live video feeds from surveillance cameras to target points of interest
- Exploited the 80% data growth in images, videos, and documents to improve customer satisfaction\*



\* Source: IBM

Big Data technology is appealing because of the following reasons:

- It helps to manage and process a huge amount of data in a cost-efficient way.
- It analyzes data in its native form, which may be unstructured, structured, or streaming.
- It captures data from fast-happening events in real time.
- It can handle failure of isolated nodes and tasks assigned to such nodes.
- It can turn data into actionable insights.



Following are the challenges that need to be addressed by Big Data technology:

## How to handle the system uptime and downtime

- Using commodity hardware for data storage and analysis
- Maintaining a copy of the same data across clusters

## How to combine data accumulated from all systems

- Analyzing data across different machines
- Merging of data

Following are some of the facts related to Hadoop and why it is required:

## What is Hadoop?

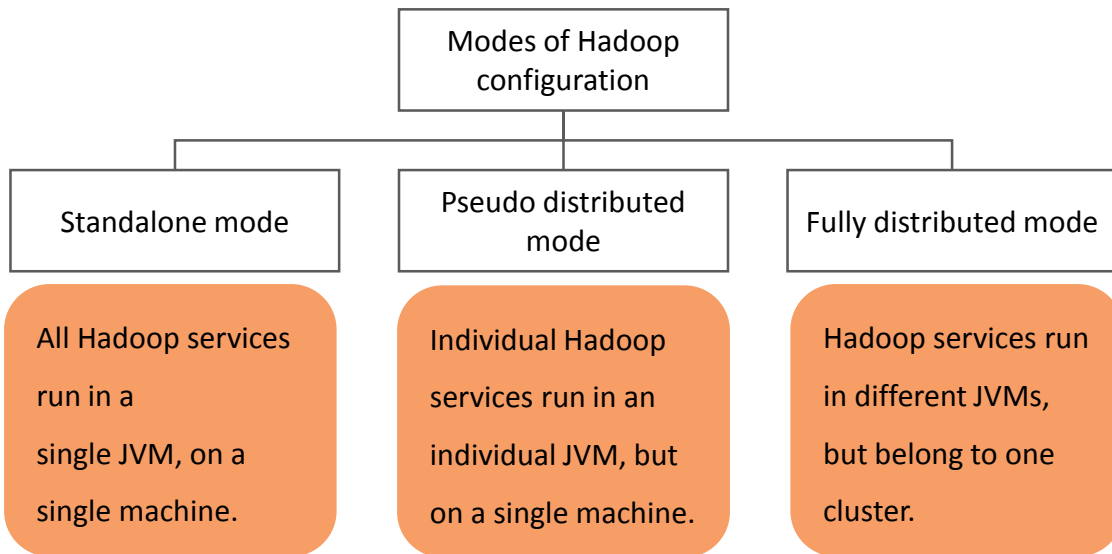
- A free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment
- Based on Google File System (GFS)

## Why Hadoop?

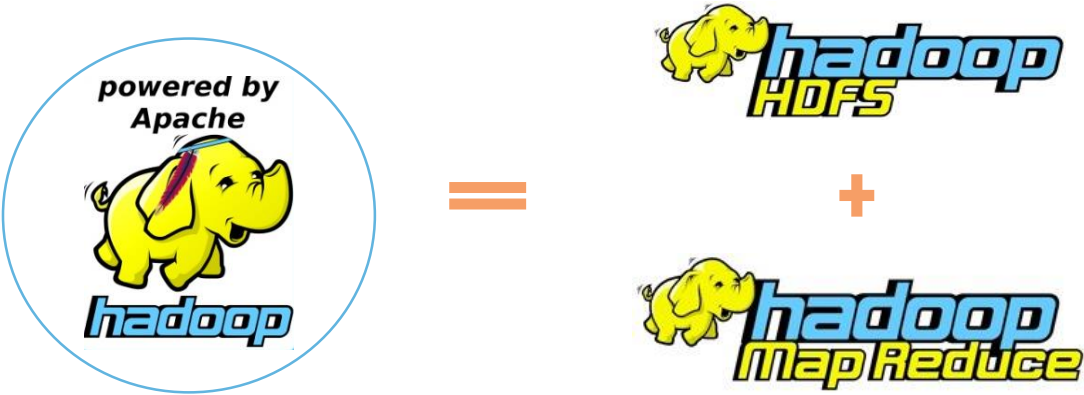
- Runs a number of applications on distributed systems with thousands of nodes involving petabytes of data
- Has a distributed file system, called Hadoop Distributed File System or HDFS, which enables fast data transfer among the nodes



Standalone, pseudo distributed, and fully distributed are the three modes of Hadoop configuration.



Hadoop HDFS and Hadoop MapReduce are the core components of Hadoop.



The key features of Hadoop HDFS are as follows:

- Provides high-throughput access to data blocks
- Provides limited interface for managing the file system to allow it to scale
- Creates multiple replicas of each data block and distributes them on computers throughout the cluster to enable reliable and rapid data access

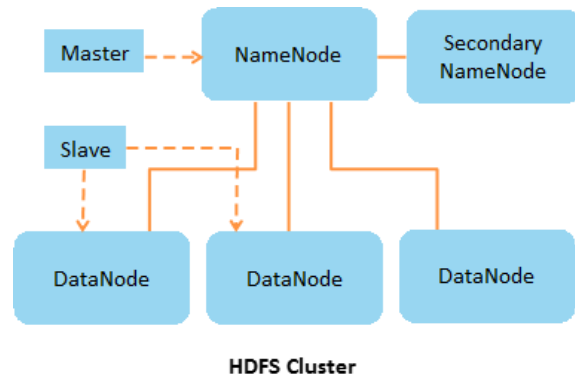
The key features of Hadoop MapReduce are as follows:

- It performs distributed data processing using the MapReduce programming paradigm.
- It allows the possession of user-defined map phase, which is a parallel, share-nothing processing of input (MapReduce paradigm).
- It allows for aggregating the output of the map phase, which is a user-defined reduce phase after a map process.

# HDFS Architecture

HDFS architecture can be summarized as follows:

- NameNode and the Secondary NameNode services constitute the master service. DataNode service is the slave service.
- The master service is responsible for accepting a job from clients and ensuring that the data required for the operation will be loaded and segregated into chunks of data blocks.
- HDFS exposes a file system namespace and allows user data to be stored in files. A file is split into one or more blocks that are stored and replicated in DataNodes. The data blocks are then distributed to the DataNode systems within the cluster. This ensures that replicas of the data are maintained.



Ubuntu is a leading open-source platform for scale-out.

Ubuntu helps in utilizing the infrastructure at its optimum level irrespective of whether users want to deploy a cloud, a web farm, or a Hadoop cluster.

Following are the benefits of Ubuntu server:

- It has the required versatility and performance to help users get the most out of the infrastructure.
- Ubuntu services ensure an efficient system administration with Landscape.
- These services provide access to Ubuntu experts as and when required, and enable fast resolution of a problem.



Following are the prerequisites for installing Hadoop:

Ubuntu Server 12.04 LTS Operating System

High-speed internet connection, for example, 512 kbps and above

Following are the prerequisites for Hadoop multi-node (Distributed mode) installation:

Ubuntu Server 12.04 VM configured in Hadoop pseudo-distributed mode

High-speed internet connection, for example, 512 kbps and above




The table shows the differences between a single-node and a multi-node cluster:


Single-node cluster	Multi-node cluster
Hadoop is installed on a single system or node.	Hadoop is installed on multiple nodes ranging from a few to thousands.
Single-node clusters are used to run trivial processes and simple MapReduce and HDFS operations. It is also used as a test bed.	Multi-node clusters are used for complex computational requirements including analytics.

MapReduce is a programming model and an associated implementation for processing and generating large data sets with parallel and distributed algorithms on a cluster.


MapReduce operation includes:




Specifying computation in terms of map and reduce functions



Parallel computation across large-scale clusters of machines



Handling machine failures and performance issues



Ensuring efficient communication between the nodes

MapReduce can be applied to significantly larger datasets when compared to "commodity" servers.

Some characteristics of MapReduce are as follows:

- It can handle very large scale data: petabytes, exabytes, and so on.
- It works well on Write Once and Read Many (WORM) data.
- It allows parallelism without mutexes.
- The Map and Reduce operations are typically performed by the same physical processor.
- The operations are provisioned near the data, that is, data locality is preferred.
- The commodity hardware and storage are leveraged.
- The runtime takes care of splitting and moving data for operations.

Some of the real-time uses of MapReduce are as follows:

- Simple algorithms such as grep, text-indexing, and reverse indexing
- Data-intensive computing such as sorting
- Data mining operations like Bayesian classification
- Search engine operations like keyword indexing, ad rendering, page rank
- Enterprise analytics
- Gaussian analysis for locating extra-terrestrial objects in astronomy
- Semantic web and web 3.0

Requirements for installing Hadoop in Ubuntu Desktop 12.04:

- Ubuntu Desktop 12.04 installed with Eclipse
- High-speed Internet connection

MapReduce functions use key/value pairs. The key features of Hadoop MapReduce are as follows:

- The framework converts each record of input into a key/value pair, which is a one-time input to the map function.
- The map output is also a set of key/value pairs which are grouped and sorted by keys.
- The reduce function is called once for each key, in sort sequence, with the key and set of values that share that key.
- The reduce method may output an arbitrary number of key/value pairs, which are written to the output files in the job output directory.

The framework provides two processes that handle the management of MapReduce jobs:

- TaskTracker manages the execution of individual map and reduce tasks on a compute node in the cluster.
- JobTracker accepts job submissions, provides job monitoring and control, and manages the distribution of tasks to the TaskTracker nodes.

The Hadoop Distributed File System (HDFS) is a block-structured, distributed file system. It is designed to run on small commodity machines in a way that the performance of running jobs will be better when compared to single standalone dedicated servers.

A diagram consisting of a large rounded rectangle with a thin black border. Inside, at the top, is a solid orange-red horizontal bar. Below this bar are three light blue rounded rectangles arranged horizontally. The text "Some of the settings in HDFS" is centered within the orange-red bar. The text "HDFS Benchmarking", "Setting up HDFS block size", and "Decommissioning a DataNode" are centered within the three blue boxes respectively.

Some of the settings in HDFS

HDFS Benchmarking

Setting up HDFS block size

Decommissioning a DataNode



Hadoop MapReduce uses data types when it works with user-given mappers and reducers. The data is read from files into mappers and emitted by mappers to reducers. The processed data is sent back by the reducers. Data emitted by reducers go into output files. At every step, data is stored in Java objects.

**Writable data types:** In the Hadoop environment, objects that can be put to or received from files and across the network must obey a particular interface called Writable.



Writable interface allows Hadoop to read and write data in a serialized form for transmission.

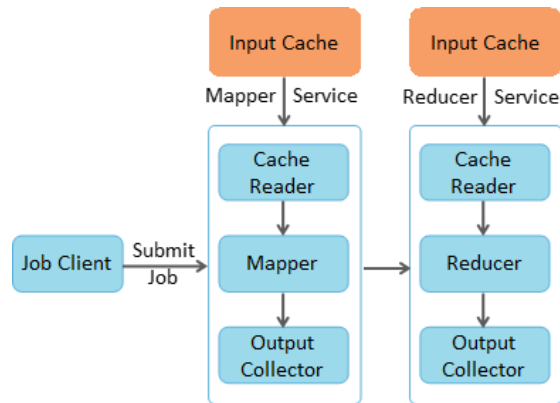
The table lists a few important data types and their functions:

Data types	Functions
Text	Stores String data
IntWritable	Stores Integer data
LongWritable	Stores Long data
FloatWritable	Stores Float data
DoubleWritable	Stores Double data
BooleanWritable	Stores Boolean data
ByteWritable	Stores Byte data
NullWritable	Placeholder when value is not needed

# Distributed Cache

Distributed Cache is a Hadoop feature that helps cache files needed by applications. Following are the functions of distributed cache:

- It helps to boost efficiency when a map or a reduce task needs access to common data.
- It lets a cluster node read the imported files from its local file system, instead of retrieving the files from other cluster nodes.
- It allows both single files and archives such as zip and tar.gz.



Following are the other functions of distributed cache:

- It copies files only to slave nodes. If there are no slave nodes in the cluster, distributed cache copies the files to the master node.
- It allows access to the cached files from mapper or reducer applications to make sure that the current working directory is added into the application path.
- It allows referencing the cached files as though they are present in the current working directory.

Joins are relational constructs that can be used to combine relations. In MapReduce, joins are applicable in situations where two or more datasets need to be combined. A join is performed either in the Map phase or in the Reduce phase by taking advantage of the MapReduce Sort-Merge architecture.

Following are the types of joins in MapReduce:

Reduce side join

Replicated join

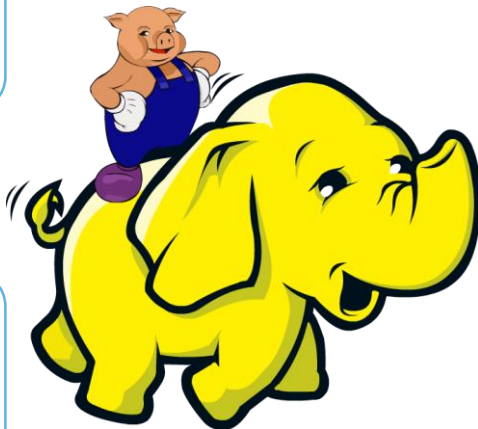
Composite join

Cartesian product

Pig is one of the components of the Hadoop eco-system.

Pig is a high-level data flow scripting language.

Pig uses HDFS for storing and retrieving data and Hadoop MapReduce for processing Big Data.



Pig runs on the Hadoop clusters.

Pig is an Apache open-source project.

Following are the major components of Pig:

## Pig Latin script language

- Procedural data flow language
- Contains syntax and commands that can be applied to implement business logic
- Example: LOAD and STORE

## Runtime engine

- Compiler that produces sequences of Map-Reduce programs
- Uses HDFS for storing and retrieving the data
- Used to interact with the Hadoop system
- Parses, validates, and compiles the script operations into a sequence of MapReduce jobs

As part of its data model, Pig supports four basic types:

## Atom

- A simple atomic value
- Example: 'Mike'

## Tuple

- A sequence of fields that can be of any data type
- Example: ('Mike', 43)

## Bag

- A collection of tuples of potentially varying structures; can contain duplicates
- Example: {('Mike'), ('Doug', (43, 45))}

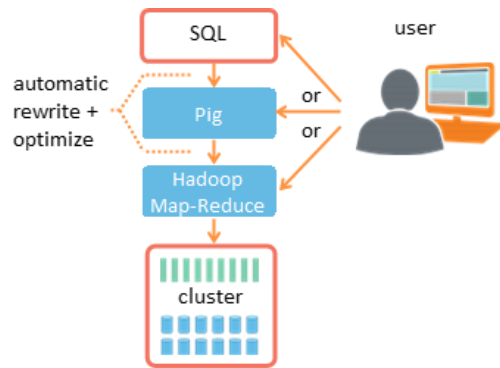
## Map

- An associative array; the key must be a chararray but the value can be any type
- Example: [name#Mike,phone#5551212]



The differences between Pig and SQL are given in the table below:

Difference	Pig	SQL
Definition	Scripting language used to interact with HDFS	Query language used to interact with databases
Query Style	Step-by-step	Single block
Evaluation	Lazy evaluation	Immediate evaluation
Pipeline Splits	Pipeline splits are supported	Requires the join to be run twice or materialized as an intermediate result



Ensure the following parameters while setting the environment for Pig Latin:

- Ensure all Hadoop services are running
- Ensure Pig is installed and configured
- Ensure all datasets are uploaded in the NameNode

Let us summarize the topics covered in this lesson:



- Big Data has three characteristics, namely, variety, velocity, and volume.
- Hadoop HDFS and Hadoop MapReduce are the core components of Hadoop.
- One of the key features of MapReduce is that the map output is a set of key/value pairs which are grouped and sorted by key.
- TaskTracker manages the execution of individual map and reduce tasks on a compute node in the cluster.
- Pig is a high-level data flow scripting language. It uses HDFS for storing and retrieving data.