SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)
MCA DEPARTMENT

LECTURE NOTES

Subject Name: BIG DATA ANALYTICS

Year / Branch: II MCA – I SEMESTER

Regulation: R24

Prepared By: Mr. N P Gangadhar,

Assistant Professor

II MCA – I SEMESTER			
COURSE CODE:	**24MCA213**	**CREDITS:**	**4**
COURSE TITLE:	**BIG DATA ANALYTICS**	**L-T-P:**	**4-0-0**

PREREQUISITES: *Courses on "Database Management Systems" , "Object Oriented Programming through JAVA" and knowledge on Intelligence Techniques.*

COURSE EDUCATIONAL OBJECTIVES*:*

CEO1 *To explore the fundamental concepts of Big Data.*
CEO2 *To Learn Basic concepts of Hadoop.*
CEO3 *To Write Hadoop MapReduce Programs for analyzing Big data.*
CEO4 *To Explore Hadoop Ecosystem.*

UNIT - I: UNDERSTANDING BIG DATA — **Lecture Hrs:10**

What is Big Data, Concepts and Terminology - Datasets, Data Analysis, Data Analytics, Big Data Characteristics – volume, velocity, variety, veracity, value, Different Types of Data – Structured Data, Unstructured Data, Semi-Structured Data, Case Study Background.

UNIT - II : HADOOP BASICS — **Lecture Hrs:10**

Brief history of Hadoop, Apache hadoop and the hadoop ecosystem. A weather dataset, analyzing the data with unix tools, analyzing the data with hadoop , Understanding different Hadoop modes, understanding Hadoop Features-Understanding HDFS, Understanding MapReduce, Learning the HDFS and Mapreduce Architecture-Understanding the HDFS architecture, Understanding the MapReduce Architecture,
Understanding the HDFS and MapReduce architecture by plot.

UNIT - III: WORKING WITH PIG AND HIVE — **Lecture Hrs:12**

Pig -Execution Types, An Example, Pig Latin-Structure, Statements,Types, Schemas, Functions, Data Processing Operators.Hive – An example, Tables –Managed Tables and External tables, Partitions and Buckets, Importing data, Altering Data, Dropping Tables, Querying Data-Sorting and Aggragating,
Mapreduce Scripts.

UNIT - IV : INTRODUCTION TO APACHE SPARK — **Lecture Hrs:12**

What is Apache Spark, Characteristics – Speed, Easy of Use, Modularity, Extensibility, Apache Spark Components as a Unified Stack, Spark Basic Data Types, Spark Core and SQL- Dynamic Partition Pruning, SQL Operations, Schemas and Creating Data Frames

UNIT - V: HBASE, ZOOKEEPER, SQOOP — **Lecture Hrs:12**

HBase Overview – Limitations of Hadoop, what is HBase,HBase and HDFS,StorageMechansim in HBase,Features of HBase, Applications of HBase. ZooKeeper Overview – what is ZooKeeper, Distributed Application, Benefits of Distributed Applications,Challenges of Distributed Applications, What is Apache
Zookeeper meant for, Benefits of ZooKeeper

TEXT BOOKS:

1. *Big Data Fundamentals: Concepts, Drivers & Techniques", 1/e, 2016, Thomas Erl, Wajid Khattak, Paul Buhler, Prentice Hall.*
2. *"Hadoop:The Definitive Guide," 3/e, 2012, Tom White, O'REILLY Publications.*
3. *"Learning Spark"*

REFERENCE BOOKS:

1. *Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics , 2012, Bill Franks, John Wiley & Sons..*

2. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, 2012, Paul C. Zikopoulos, Chris Eaton, Dirk deRoos, Thomas Deutsch, George Lapis, McGraw-Hill.*

3. *Intelligent Data Analysis, 2007, Michael Berthhold, David J.Hand, Springer.*

4. *Understanding Big Data: Analytics for Enterprise Class Hadoop and streaming data, 2011, Paul Zikopoulos, Chris Eaton, Paul Zikopoulos, McGraw hill.*

5. *Big Data for Dummies, 2012, Hurwitz, Alan Nugent, Dr. Fern Halper, Marcia Kaufman,John Wiley & Sons*

COURSE OUTCOMES: *On successful completion of this course, students will be able to:*		POs related to COs
CO1	**Realize** characteristics of Big Data and various types of data like structured, unstructured and semistructured	PO1,PO2
CO2	**Understand** two major components of Hadoop	PO1,PO2,PO3,PO4, PO5
CO3	**Analyze** Big data using Hadoop Map Reduce programs	PO1, PO2,PO4, PO5
CO4	get **Acquainted** with two Data Access Components of Hadoop Ecosystem called pig and hive	PO1,PO3, PO4
CO5	Acquire **Knowledge** on Data Storage Component of Hadoop Ecosystem called Hbase, Data Integration Component of Hadoop Ecosystem like sqoop and Monitoring, Management and Orchestration of Hadoop Ecosystem component like zookeeper	PO1,PO2, PO3,PO4

CO-PO MAPPING(DETAILED; HIGH:3; MEDIUM:2; LOW:1)

Course	**POs** / **COs**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
C303: Big Data Analytics	**C303.1**	**3**	**2**	-	-	-	-	-	-
	C303.2	**3**	**3**	**2**	**3**	**2**	-	-	-
	C303.3	**3**	**3**	-	**3**	**3**	-	-	-
	C303.4	**3**	-	**3**	**3**	-	-	-	-
	C303.5	**3**	**3**	**2**	**3**	-	-	-	-
	C303	**3**	**2.75**	**2.3**	**3**	**2.5**	-	-	-

UNIT V - HBASE, ZOOKEEPER, SQOOP

"Big Data succeeds with fast storage from HBase and dependable coordination from Zookeeper."

Unit Overview

This unit introduces **Apache HBase** and **Apache ZooKeeper**, two important components of the Hadoop ecosystem. Students will learn the limitations of Hadoop HDFS, the need for HBase, HBase architecture, storage mechanism, features, and real-world applications. The unit also covers Apache ZooKeeper, distributed applications, benefits and challenges of distributed systems, the purpose of ZooKeeper, and how it provides coordination, synchronization, and reliability for distributed applications.

Learning Outcomes

After completing this unit, students will be able to:

1. Explain the limitations of Hadoop HDFS.
2. Describe Apache HBase and its architecture.
3. Explain the relationship between HBase and HDFS.
4. Understand the storage mechanism of HBase.
5. Describe the features and applications of HBase.
6. Explain distributed applications and their importance.
7. Describe Apache ZooKeeper and its architecture.
8. Identify the benefits and challenges of distributed applications.
9. Explain the role of ZooKeeper in distributed systems.
10. Apply HBase and ZooKeeper concepts in real-world Big Data environments.

Importance of Studying this Unit

Hadoop HDFS is excellent for storing large files, but it is not suitable for applications requiring fast, random read/write operations. Apache HBase addresses this limitation by providing a distributed NoSQL database on top of HDFS. Apache ZooKeeper ensures coordination among distributed applications by managing configuration, synchronization, leader election, and fault tolerance. Together, HBase and ZooKeeper are widely used in cloud computing, social media, banking, healthcare, and e-commerce applications.

Key Concepts

HBase, HDFS, NoSQL Database, Column-Oriented Database, Region Server, HMaster, MemStore, HFile, Write-Ahead Log (WAL), ZooKeeper, Distributed Applications, Leader Election, Synchronization, Fault Tolerance, Configuration Management, Cluster Coordination.

HBase Overview

- ❖ HBase is a
- ✓ **Non-Relational Database Management System**
- ✓ **Column-Oriented** that runs on top of HDFS

Database are classified into 2 types

1. **Relational Database**
2. **Non-Relational Database**

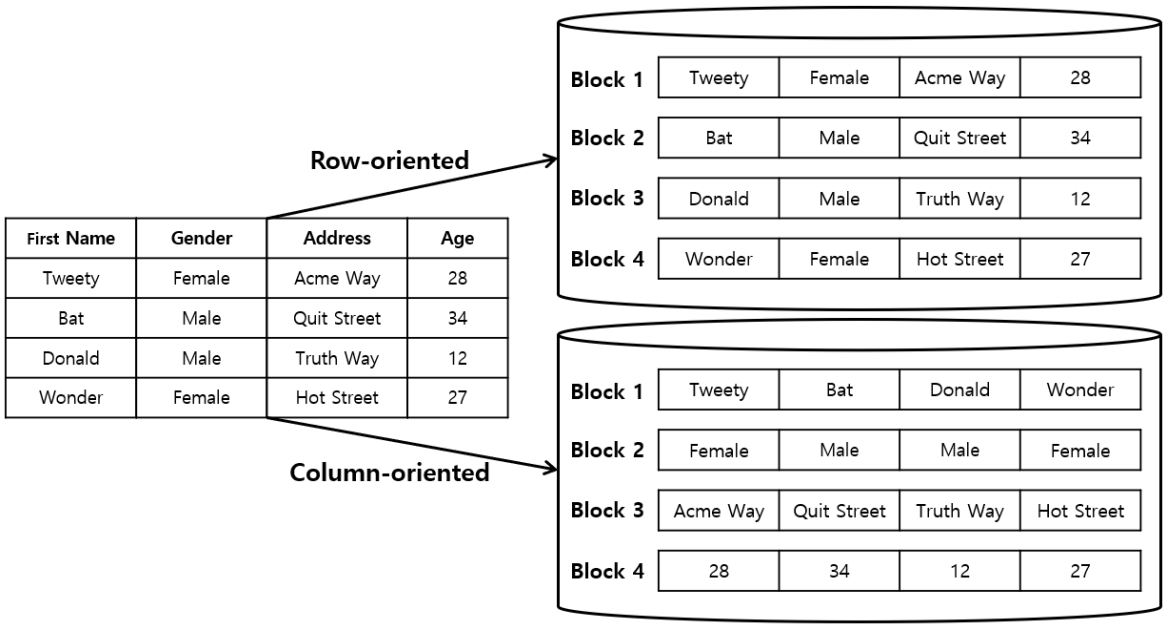
Relational Database	**Non-Relational Database**
A relational database stores data in tables composed of rows and columns.	A non-relational database is a type of database that doesn't store data in tables but instead in whatever format is best for the type of data being stored.
Relational databases are suitable for storing, retrieving, and manipulating well-defined**, structured data.**	used to store a mix of structured and unstructured data.
Some of the most common relational databases include: MySQL, IBM Db2, Snowflake, Amazon ,Aurora,,PostgreSQL, Microsoft SQL Server	Some of the most common non-relational databases include: MongoDB, ,IBM Cloundant, Amazon DynamoDB, Apache Cassandra , HBase
SQL, or Structured Query Language, is the most common programming language used to interface with relational databases	Non-relational databases are said to be NoSQL, meaning that they don't use Structured Query Language, even though many NoSQL databases do support SQL queries.

Data store are classified into 2 types

1) Row Oriented Data store or Row Oriented Database
2) Column Oriented store or Column Oriented Database

Row Oriented Database	Column Oriented Database
Row Oriented Databases are Databases that Organize data by record, keeping all of the data associated with a record next to each other in memory.	Column Oriented Databases are Databases that Organize data by field, keeping all of the data associated with a field next to each other in memory.
In a row store, or row oriented database, the data is stored row by row, such that the first column of a row will be next to the last column of the previous row.	In a Column store, or Column oriented database, the data is stored field by field
They are Optimized for reading and writing rows effectively	They are Optimized for reading and writing Columns effectively
If an application needs row wise data for processing, then better to choose row oriented database like oracle, MySQL etc..	If an application needs Column wise data for processing, then better to choose column oriented database like Hbase,Cassandra etc..
Commonly used for OLTP style applications	Commonly used for OLAP style applications

Row vs Column Oriented Databases



Row Oriented Database

100	**Ram**	**20000**	**101**	**Sita**	**25000**	**102**	**Lak**	**20000**

Column Oriented Database

100	**101**	**102**	**103**	**Ram**	**Siata**	**Lak**	**Bharath**	**20000**

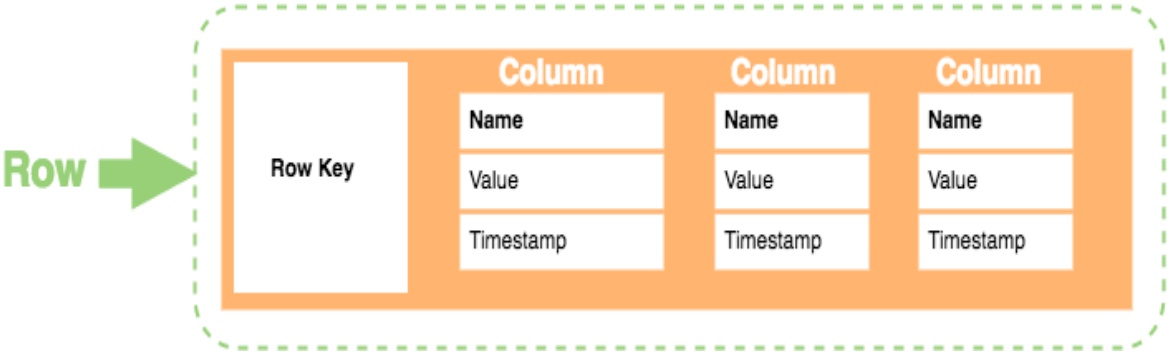
Limitations of Hadoop (or) HBase vs HDFS

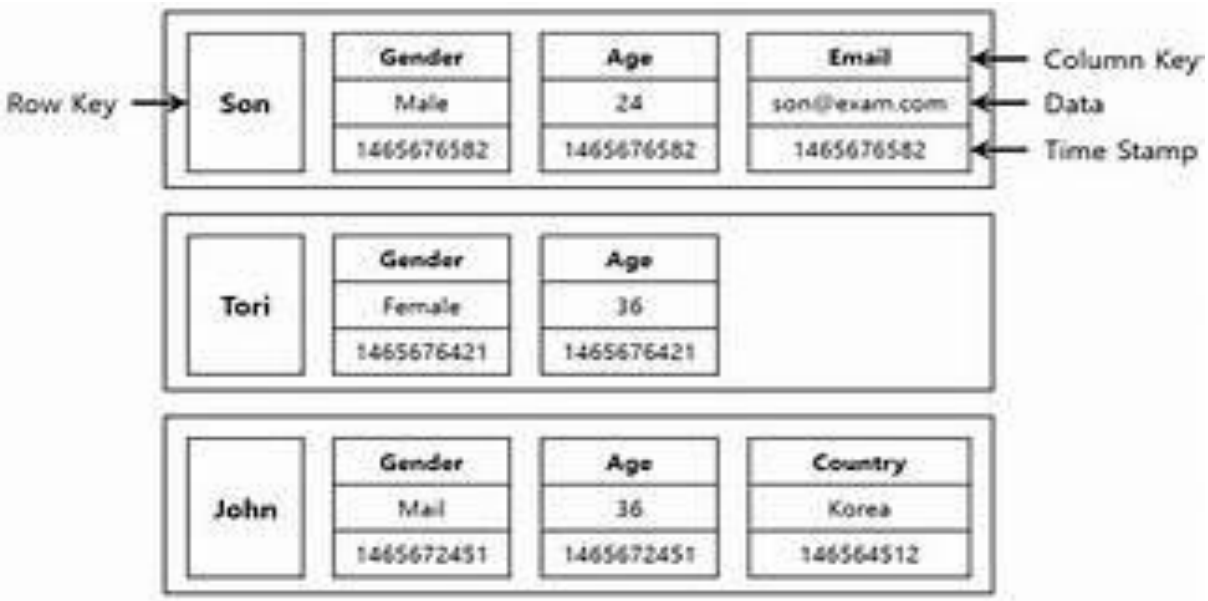
HDFS	HBase
HDFS performs Batch Processing **Batch Processing** : The Collection and Storage of Data, for processing at a scheduled time when a sufficient amount of data has been accumulated **Example :** Credit Card Bill Generation	**HBase performs Real Time Processing** **Real Time Processing** : The Immediate Processing of Data after the transaction occurs with the Database being updated at the time of the event **Example :** Flight Reservation processing
HDFS does sequential access of Data Means one has to search the entire dataset even for the simplest of Jobs	**Hbase does Random access of Data** Means one can access data randomly

Storage Mechanism of HBase

- HBase is a column-oriented database and the tables in it are sorted by row.
- The table schema defines only column families, which are the key value pairs.
- A table have multiple column families and each column family can have any number of columns.
- Subsequent column values are stored contiguously on the disk.
- Each cell value of the table has a timestamp.
- In short, in an HBase:
- Table is a collection of rows.
- Row is a collection of column families.

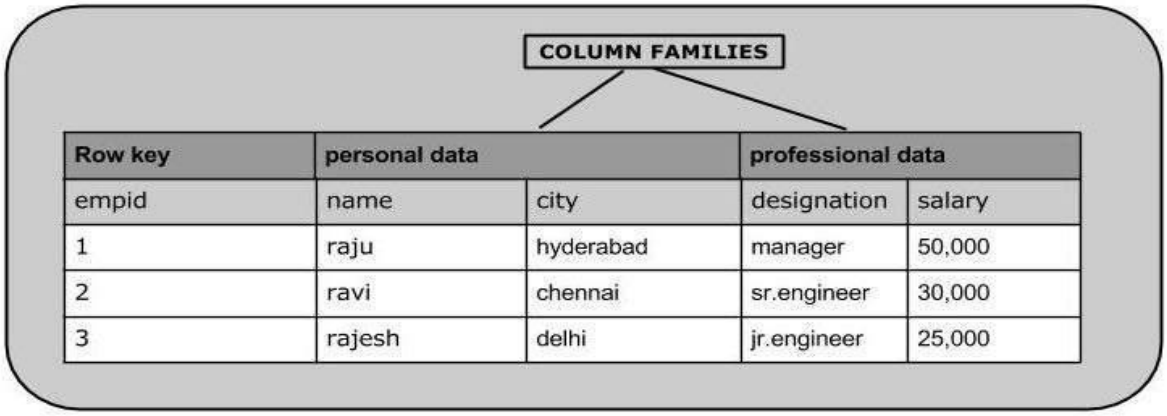
- Column family is a collection of columns.
- Column is a collection of key value pairs.
- HBase uses a column-oriented storage mechanism, and its architecture is inspired by Google's Bigtable.
- The storage mechanism is designed to provide high performance, scalability, and fault tolerance.
- Data in HBase is organized into column families.
- A column family is a logical grouping of columns, and all the columns within a column family are stored together in an HFile.
- Each column family can have a different set of columns.





Given below is an example schema of table in HBase.

Rowid	Column Family			Column Family			Column Family			Column Family		
	col1	col2	col3	col1	col2	col3	col1	col2	col3	col1	col2	col3
1												
2												
3												



Row key	personal data		professional data	
empid	name	city	designation	salary
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

Features of HBase

- HBase is a Column oriented Database which is commonly used for OLAP Applications.
- Hbase performs Real Time Processing
- HBase is a Open source Data warehouse component built on top of Hadoop
- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and writes.
- It integrates with Hadoop, both as a source and a destination.
- It has easy java API for client.
- It provides data replication across clusters.

Applications of HBase

- Apache HBase is used to have random, real-time read/write access to Big Data.
- It hosts very large tables on top of clusters of commodity hardware.
- Apache HBase is a non-relational database modeled after Google's Bigtable.
- Bigtable acts up on Google File System, likewise Apache HBase works on top of Hadoop and HDFS.
- It is used whenever there is a need to write heavy applications.
- HBase is used whenever we need to provide fast random access to available data.
- Companies such as Facebook, Twitter, Yahoo, and Adobe use HBase internally.

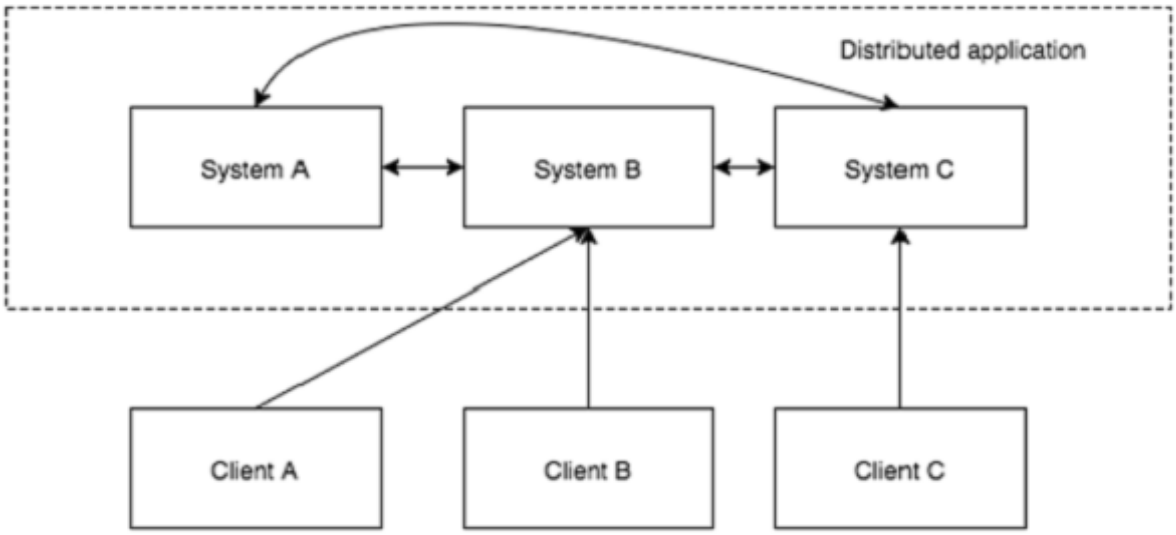
WHAT IS ZOOKEEPER

- ZooKeeper is a distributed co-ordination service to manage large set of hosts.
- Co-ordinating and managing a service in a distributed environment is a complicated process.
- ZooKeeper solves this issue with its simple architecture and API.
- ZooKeeper allows developers to focus on core application logic without worrying about the distributed nature of the application.
- The ZooKeeper framework was originally built at "Yahoo!" for accessing their applications in an easy and robust manner.
- Later, Apache ZooKeeper became a standard for organized service used by Hadoop, HBase, and other distributed frameworks.
- For example, Apache HBase uses ZooKeeper to track the status of distributed data.

DISTRIBUTED APPLICATIONS

- A distributed application can run on multiple systems in a network at a given time (simultaneously) by coordinating among themselves to complete a particular task in a fast and efficient manner.
- Normally, complex and time-consuming tasks, which will take hours to complete by a non-distributed application (running in a single system) can be done in minutes by a distributed application by using computing capabilities of all the system involved
- The time to complete the task can be further reduced by configuring the distributed application to run on more systems.

- A group of systems in which a distributed application is running is called a Cluster and each machine running in a cluster is called a Node.
- A distributed application has two parts, Server and Client application.
- Server applications are actually distributed and have a common interface so that clients can connect to any server in the cluster and get the same result.
- Client applications are the tools to interact with a distributed application



Benefits of Distributed Applications

- **Reliability** − Failure of a single or a few systems does not make the whole system to fail.
- **Scalability** − Performance can be increased as and when needed by adding more machines with minor change in the configuration of the application with no downtime.
- **Transparency** − Hides the complexity of the system and shows itself as a single entity / application.

Challenges of Distributed Applications

- **Race condition** − Two or more machines trying to perform a particular task, which actually needs to be done only by a single machine at any given time. For example, shared resources should only be modified by a single machine at any given time.
- **Deadlock** − Two or more operations waiting for each other to complete indefinitely.
- **Inconsistency** − Partial failure of data.

What is Apache ZooKeeper Meant For?

- Apache ZooKeeper is a service used by a cluster (group of nodes) to coordinate between themselves and maintain shared data with robust synchronization techniques. ZooKeeper is itself a distributed application providing services for writing a distributed application.
- The common services provided by ZooKeeper are as follows −
  - **Naming service** − Identifying the nodes in a cluster by name. It is similar to DNS, but for nodes.
  - **Configuration management** − Latest and up-to-date configuration information of the system for a joining node.
  - **Cluster management** − Joining / leaving of a node in a cluster and node status at real time.
  - **Leader election** − Electing a node as leader for coordination purpose.
  - **Locking and synchronization service** − Locking the data while modifying it. This mechanism helps you in automatic fail recovery while connecting other distributed applications like Apache HBase.
  - **Highly reliable data registry** − Availability of data even when one or a few nodes are down.
- Distributed applications offer a lot of benefits, but they throw a few complex and hard-to-crack challenges as well.
- ZooKeeper framework provides a complete mechanism to overcome all the challenges.
- Race condition and deadlock are handled using fail-safe synchronization approach.
- Another main drawback is inconsistency of data, which ZooKeeper resolves with atomicity.

Benefits of ZooKeeper

- Here are the benefits of using ZooKeeper −
- Simple distributed coordination process
- Synchronization − Mutual exclusion and co-operation between server processes. This process helps in Apache HBase for configuration management.
- Ordered Messages
- Serialization − Encode the data according to specific rules. Ensure your application runs consistently. This approach can be used in MapReduce to coordinate queue to execute running threads.
- Reliability

- Atomicity − Data transfer either succeed or fail completely, but no transaction is partial

Unit Highlights

- Hadoop HDFS is designed for batch processing but has limitations for real-time read/write operations.
- Apache HBase is a distributed, column-oriented NoSQL database built on HDFS.
- HBase uses components such as HMaster, Region Server, MemStore, WAL, and HFiles for efficient storage and retrieval.
- HBase supports real-time access, horizontal scalability, and fault tolerance.
- Apache ZooKeeper is a distributed coordination service that manages configuration, synchronization, and leader election.
- Distributed applications offer scalability and reliability but face challenges such as synchronization, consistency, and fault management.
- ZooKeeper simplifies distributed system management by providing centralized coordination and automatic recovery.
- HBase and ZooKeeper are widely used in social media, banking, healthcare, IoT, telecommunications, and cloud computing for building reliable Big Data applications.

TEXT BOOKS:

1. Big Data Fundamentals: Concepts, Drivers & Techniques", 1/e, 2016, Thomas Erl, Wajid Khattak, Paul Buhler, Prentice Hall.
2. "Hadoop:The Definitive Guide," 3/e, 2012, Tom White, O'REILLY Publications.
3. "Learning Spark"

REFERENCE BOOKS:

1. Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics
   , 2012, Bill Franks, John Wiley & Sons..
2. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, 2012, Paul C. Zikopoulos, Chris Eaton, Dirk deRoos, Thomas Deutsch, George Lapis, McGraw-Hill.
3. Intelligent Data Analysis, 2007, Michael Berthhold, David J.Hand, Springer.
4. Understanding Big Data: Analytics for Enterprise Class Hadoop and

streaming data, 2011, Paul Zikopoulos, Chris Eaton, Paul Zikopoulos, McGraw hill.

5. Big Data for Dummies, 2012, Hurwitz, Alan Nugent, Dr. Fern Halper, Marcia Kaufman,John Wiley & Sons

S.No	Questions
UNIT V – HBase, Zookeeper	
PART-A (Two Marks Questions)	
1.	What is Hbase
2.	What is zookeeper
3.	What is sqoop
4.	List out the limitations of Hadoop
5.	Mention the features of HBase
6.	Define distributed application
7.	Mention the challenges of distributed applications
8.	Differentiate Hbase and HDFS
9.	List out the Benefits of Zookeeper
PART-B (Ten Marks Questions)	
1.	Discuss the Limitations of Hadoop and how they are overcomed with HBase
2.	Write about the Hbase and HDFS in detail
3.	Describe the storage mechanism of Hbase in detail
4.	Explain the features and Applications of HBase
5.	Write about distributed application and benefits of distributed applications
6.	What is apache zookeeper meant for
7.	Describe the benefits and challenges of zookeeper
8.	Write about limitations, applications and storage mechanism of Apache hbase