BIG DATA

Syllabus

Session: 1, 2 & 3

Introduction to Big Data

o Big Data - Beyond the Hype,

o Big Data Skills and Sources of Big Data,

o Big Data Adoption,

o Research and Changing Nature of Data Repositories,

o Data Sharing and Reuse Practices and Their Implications for Repository Data Curation,

o Overlooked and Overrated Data Sharing,

Data sharing can be overlooked if organizations or individuals do not see the value in sharing their data or if they are not aware of the benefits of data sharing. This can lead to missed opportunities for collaboration and innovation.

Data sharing can also be overrated if the focus is placed too heavily on the act of sharing data, rather than on the value that can be derived from the data. This can lead to a lack of attention on important issues such as data privacy and security.

o Data Curation Services in Action,

o Open Exit: Reaching the End of The Data Life Cycle,

o The Current State of Meta-Repositories for Data

o Curation of Scientific Data at Risk of Loss: Data Rescue And Dissemination

Introduction to Hadoop

o A Brief History of Hadoop,

o Evolution of Hadoop,

o Introduction to Hadoop and its components

0 Comparison with Other Systems,

o Hadoop Releases

o Hadoop Distributions and Vendors

Hadoop is an open-source software framework that allows for the distributed processing of large data sets across clusters of computers. Hadoop Distributions are the pre-configured and packaged version of Hadoop which makes it easy to install and manage.

Examples of popular Hadoop distributions include Apache Hadoop, Cloudera, Hortonworks, MapR, etc.

Hadoop Vendors are the companies that provide Hadoop distributions, support and services. These vendors often offer additional tools and features to the open-source version of Hadoop, such as improved security, enhanced scalability, and additional management features.

Some of the popular Hadoop vendors include Cloudera, Hortonworks, MapR, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.

Session: 4 & 5

Hadoop Distributed File System (HDFS)

o Distributed File System,

o What is HDFS,

Hadoop Distributed File System (HDFS) is a distributed file system that is part of the Hadoop ecosystem. It is designed to store and manage large data sets across a cluster of commodity computers. HDFS is built on the principle of storing large files across multiple machines, rather than storing them on a single machine, which allows for improved scalability and fault tolerance.

o Where does HDFS fit in,

Hadoop Distributed File System (HDFS) is an integral component of the Hadoop ecosystem and it fits in as a storage layer for big data processing tasks.

o Core components of HDFS,

Hadoop Distributed File System (HDFS) has a master-slave architecture, with two core components: the NameNode and the DataNode.

1. Use of Secondary namenode

The fsimage contains a snapshot of the entire HDFS namespace at a certain point in time, while the edit logs contain a record of changes made to the file system since the last fsimage was created.

The SNN merges these files to create a new fsimage, which is then used by the NameNode to start up.

it is not a backup of the NameNode.

It helps in monitoring the health and status of the HDFS cluster

o HDFS Daemons,

In Hadoop Distributed File System (HDFS), there are two main daemons that run on each node of the cluster: the NameNode daemon and the DataNode daemon.

o Hadoop Server Roles: Name Node, Secondary Name Node, and Data Node HDFS Architecture

o HDFS Architecture,

o Scaling and Rebalancing,

Scaling and Rebalancing are two important concepts in Hadoop Distributed File System (HDFS) and big data processing in general.

Scaling: Scaling refers to the process of adding more resources to a cluster in order to handle an increase in data volume, velocity, or variety. In HDFS, scaling can be achieved by adding more DataNodes to the cluster in order to increase storage capacity and processing power. Scaling can also be achieved by adding more NameNodes to handle increased load on the master node.

Rebalancing: Rebalancing refers to the process of redistributing data blocks across the DataNodes in a cluster in order to ensure that the cluster is operating at optimal capacity. This can be done to account for the addition or removal of DataNodes, or to ensure that the data is evenly distributed across the DataNodes.

o Replication,

Replication: Replication in HDFS refers to the process of creating multiple copies of data blocks and storing them across different DataNodes in the cluster. This is done to ensure that if a DataNode fails, the data can still be accessed from other replicas. The replication factor, which is the number of copies of a data block that are stored in the cluster, can be configured to suit the needs of the application.

o Rack Awareness,

Rack Awareness: Rack Awareness is a feature of HDFS that allows for data blocks to be stored in such a way that they are spread across different racks in a cluster, rather than being stored on the same rack. This is done to ensure that if a rack fails, the data can still be accessed from other racks.

o Data Pipelining,

Data Pipelining: Data pipelining refers to the process of performing data processing tasks on data as it is being transferred between nodes, rather than waiting for the data to be completely transferred before processing it. This can be done to improve the performance of data processing tasks and to reduce the amount of data that needs to be stored in the cluster.

o Node Failure Management.

Node Failure Management: Node failure management refers to the process of detecting and handling the failure of a node in the cluster. HDFS uses a combination of heartbeats and block reports to detect when a node has failed, and it uses replication and rack awareness to ensure that data is still accessible even if a node fails.

o HDFS High Availability NameNode

HDFS High Availability NameNode: HDFS High Availability (HA) refers to the process of having two or more active NameNodes in a cluster, with one acting as a primary NameNode and the other as a standby NameNode. In case of primary NameNode failure, the standby NameNode takes over, providing a highly available and fault-tolerant HDFS service.

Hadoop Installation and Cluster Configuration

Session: 6

Getting Started: Hadoop Installation

o Hadoop Operation modes

There are three main operation modes

These are the three main operation modes of Hadoop, and the choice of mode depends on the specific use case and requirements of the organization

.Standalone mode is suitable for testing and development,

Pseudo-Distributed mode is useful for small scale testing and development,

Fully-Distributed mode is suitable for production use and large scale data processing.

o Setting up a Hadoop Cluster,

o Cluster specification,

o Single and Multi-Node Cluster Setup on Virtual & Physical Machines,

o Remote Login using Putty/Mac Terminal/Ubuntu Terminal.

0 Hadoop Configuration, Security in Hadoop, Administering Hadoop,

o HDFS — Monitoring & Maintenance, Hadoop benchmarks,

o Hadoop in the cloud.

Session: 7

Hadoop Architecture

o Hadoop Architecture,

o Core components of Hadoop,

o Common Hadoop Shell commands.

Session: 8

HDFS Data Storage Process

o HDFS Data storage process,

o Anatomy of writing and reading file in HDFS,

o Handling Read/Write failures

o HDFS user and admin commands,

o HDFS Web Interface.

Map Reduce Session: 9

Getting in touch with Map Reduce Framework

o Hadoop Map Reduce paradigm,

o Map and Reduce tasks,

o Map Reduce Execution Framework,

o Map Reduce Daemons

o Anatomy of a Map Reduce Job run

More Map Reduce Concepts

o Partitioners and Combiners,

o Input Formats (Input Splits and Records, Text Input, Binary Input, Multiple Inputs),

o Output Formats (Text Output, Binary Output, Multiple Output).

o Distributed Cache

Session: 10

Basics of Map Reduce Programming

o Hadoop Data Types,

o Java and Map Reduce,

o Map Reduce program structure,

o Map-only program, Reduce-only program,

o Use of combiner and partitioner,

o Counters, Schedulers (Job Scheduling),

o Custom Writables, Compression

Session: 11

Map Reduce Streaming

o Complex Map Reduce programming,

o Map Reduce streaming,

o Python and Map Reduce,

o Map Reduce on image dataset

Hadoop ETL

Session: 12

o Hadoop ETL Development,

o ETL Process in Hadoop,

o Discussion of ETL functions,

o Data Extractions,

o Need of ETL tools,

o Advantages of ETL tools.

HBase

Session: 13

Introduction to HBase

o Overview of HBase

o HBase architecture

o Installation

Session: 14 and 15

The HBaseAdmin and HBase Security

o Various Operations on Tables

o HBase general command and shell,

o java client API for HBase

o Admin API

o CRUD operations

o Client API

o HBase — Scan, Count and Truncate

o HBase Security

Hive

Session: 16

The Hive Data-ware House

o Introduction to Hive,

o Hive architecture and Installation,

o Comparison with Traditional Database,

o Basics of Hive Query Language.

Session: 17

Working with Hive QL

o Datatypes,

o Operators and Functions,

o Hive Tables (Managed Tables and Extended Tables),

o Partitions and Buckets,

o Storage Formats,

o Importing data,

o Altering and Dropping Tables

Session:18

Querying with Hive QL

o Querying Data-Sorting,

o Aggregating,

o Map Reduce Scripts,

o Joins and Sub queries,

o Views,

o Map and Reduce side joins to optimize query.

Session: 19

More on Hive QL

o Data manipulation with Hive,

o UDFs,

o Appending data into existing Hive table,

o custom map/reduce in Hive

o Writing HQL scripts

Apache Airflow

Session: 20, 21and 22

o Introduction to Data Warehousing and Data Lakes

o Designing Data warehousing for an ETL Data Pipeline

o Designing Data Lakes for an ETL Data Pipeline

o ETL vs ELT

o Fundamentals of Airflow

o Work management with Airflow

o Automating an entire Data Pipeline with Airflow

Introduction to Apache Spark& Kafka

Session: 23, 24 and 25

Apache Spark APIs for large-scale data processing

o Overview, Linking with Spark, Initializing Spark,

o Resilient Distributed Datasets (RDDs), External Datasets

o RDD v/s Data frames v/s Datasets

o Data frame operations

o Structured Spark Streaming

o Passing Functions to Spark, Working with Key-Value Pairs, Shuffle operations,

o RDD Persistence, Removing Data, Shared Variables, Deploying to a Cluster

Session: 26

o Map Reduce with Spark

o Working with Spark with Hadoop

0 Working with Spark without Hadoop and their Differences

Session: 27

o Data preprocessing

o EDA

Session: 28 and 29

o Introduction to Kafka

o Working with Kafka using Spark

o Spark streaming Architecture

o Spark Streaming APIs

o Building Stream Processing Application with Spark

Session: 30

o Setting up Kafka Producer and Consumer

o Kafka Connect API

Session: 31

o Spark SQL

Session: 32 and 33

o Spark MLIib

o Predictive Analysis