Introduction to Big Data

DECAP456

Edited by
Sartaj Singh

LOVELY PROFESSIONAL UNIVERSITY

Introduction to Big Data

Edited By:
Sartaj Singh

Title: INTRODUCTION TO BIG DATA

Author's Name: Dr. Rajni Bhalla

Published By : Lovely Professional University

Publisher Address: Lovely Professional University, Jalandhar Delhi GT road, Phagwara - 144411

Printer Detail: Lovely Professional University

Edition Detail: (I)

ISBN: 978-93-94068-26-1

CONTENT

Unit 01: Introduction to Big Data

CONTENTS

Objectives

After studying this unit, you will be able to:

- understand what is BIG DATA.
- understand Applications of BIG DATA
- learn tools used in BIG DATA
- known challenges in BIG DATA

Introduction

The quantity of data created by humans is quickly increasing every year as a result of the introduction of new technology, gadgets, and communication channels such as social networking sites.Big data is a group of enormous datasets that can't be handled with typical computer methods. It is no longer a single technique or tool; rather, it has evolved into a comprehensive subject including a variety of tools, techniques, and frameworks.Quantities, letters, or symbols on which a computer performs operations and which can be stored and communicated as electrical signals and recorded on magnetic, optical, or mechanical media.

1.1 What is Big Data

Big Data is a massive collection of data that continues to increase dramatically over time. It is a data set that is so huge and complicated that no typical data management technologies can effectively store or process it. Big data is similar to regular data, except it is much larger.Big data analytics is the use of advanced analytic techniques to very large, heterogeneous data sets, which can contain structured, semi-structured, and unstructured data, as well as data from many sources and sizes ranging from terabytes to zettabytes.

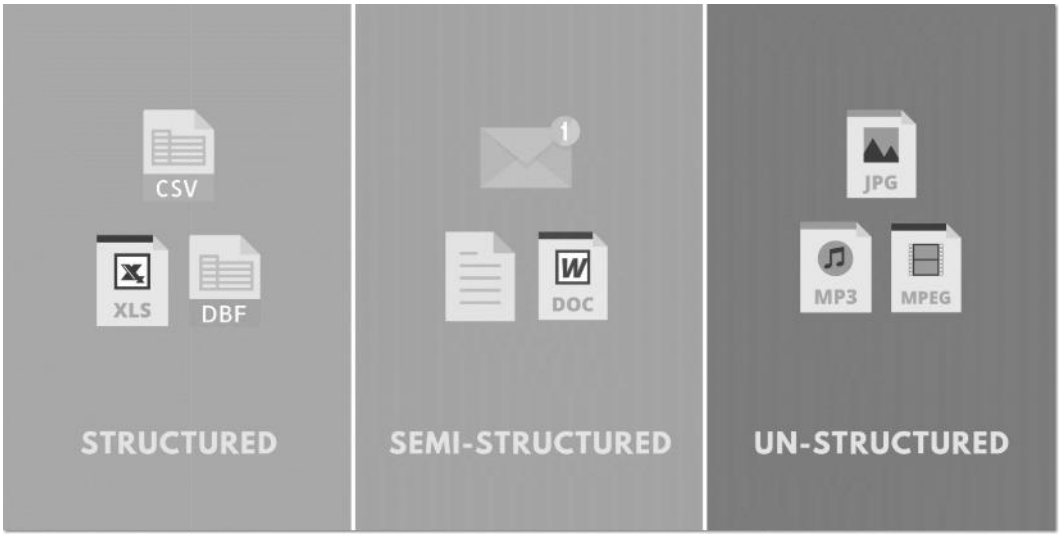


Figure 1 Structured,Semi-structured and Un-structured

Big data is a term that defines the massive amount of organized and unstructured data that a company encounters on a daily basis.

Note

- It may be studied for insights that lead to improved business choices and strategic movements.
- It is a collection of organized, semi-structured, and unstructured data that may be mined for information and utilized in machine learning, predictive modelling, and other advanced analytics initiatives.

Examples of Big Data

Figure 2 shows an example of big data. Every day, 500+ terabytes of fresh data are absorbed into the Facebook systems. This information is mostly gathered through photo and video uploads, message exchanges, and the posting of comments, among other things.

In 30 minutes of flying time, a single Jet engine may create 10+ gigabytes of data. With thousands of flights every day, the amount of data generated can amount to several Petabytes.Every day, the Fresh York Stock Exchange creates around a terabyte of new trading data.

Figure 2: Example of Big Data

1.2 Characteristics of Big Data

Big data can be described by following characteristics as shown in Figure 3.

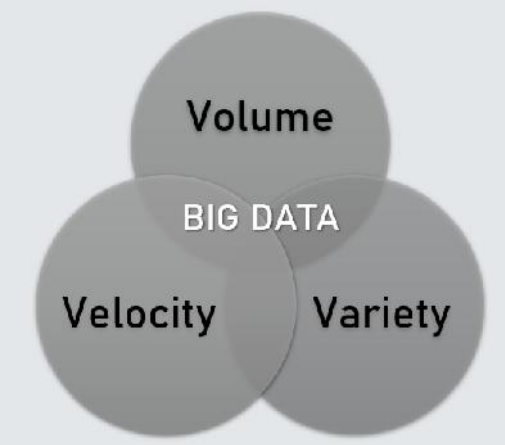


Figure 3 Characteristics of Big Data

Volume

The term 'Big Data' refers to a massive amount of information. The term "volume" refers to a large amount of data. The magnitude of data plays a critical role in determining its worth. When the amount of data is extremely vast, it is referred to as 'Big Data.'

This means that the volume of data determines whether or not a set of data may be classified as Big Data. As a result, while dealing with Big Data, it is vital to consider a certain 'Volume.'

Example:

In 2016, worldwide mobile traffic was predicted to be 6.2 Exabytes (6.2 billion GB) per month. Furthermore, by 2020, we will have about 40000 ExaBytes of data.

Velocity

The term "velocity" refers to the rapid collection of data. Data comes in at a high rate from machines, networks, social media, mobile phones, and other sources in Big Data velocity. A large and constant influx of data exists. This influences the data's potential, or how quickly data is created and processed in order to satisfy needs. Data sampling can assist in dealing with issues such as'velocity.' For instance, Google receives more than 3.5 billion queries every day. In addition, the number of Facebook users is growing at a rate of around 22% every year.

Variety

Structured data is just data that has been arranged. It usually refers to data that has been specified in terms of length and format.

Semi-structured data is a type of data that is semi-organized. It's a type of data that doesn't follow the traditional data structure. This sort of data is represented by log files.

Unstructured data is just data that has not been arranged. It usually refers to data that doesn't fit cleanly into a relational database's standard row and column structure.Texts, pictures, videos etc. are the examples of unstructured data which can't be stored in the form of rows and columns.

Benefits of Big Data Processing

Ability to process Big Data brings in multiple benefits, such as-

1. Businesses can utilize outside intelligence while taking decisions.
2. Access to social data from search engines and sites like facebook, twitter are enabling organizations to fine tune their business strategies.

3. Improved customer service (Traditional customer feedback systems are getting replaced by new systems designed with Big Data technologies.
4. Improved customer service (In these new systems, Big Data and natural language processing technologies are being used to read and evaluate consumer responses.
5. Early identification of risk to the product/services, if any
6. Better operational efficiency

Big Data technologies can be used for creating a staging area or landing zone for new data before identifying what data should be moved to the data warehouse. In addition, such integration of Big Data technologies and data warehouse helps an organization to offload infrequently accessed data.

Why is Big Data Important?

- ***Cost Savings***

  Big data helps in providing business intelligence that can reduce **costs** and improve the efficiency of operations. Processes like quality assurance and testing can involve many complications particularly in industries like biopharmaceuticals and nanotechnologies

- **Time Reductions**

  Companies may collect data from a variety of sources using real-time in-memory analytics. Tools like Hadoop enable businesses to evaluate data quickly, allowing them to make swift decisions based on their findings.

- **Understand the market conditions**

  Businesses can benefit from big data analysis by gaining a better grasp of market conditions.

  Analysing client purchase behaviour, for example, enables businesses to discover the most popular items and develop them appropriately. This allows businesses to stay ahead of the competition.

- **Social Media Listening's**

  Companies can perform sentiment analysis using Big Data tools. These enable them to get feedback about their company, that is, who is saying what about the company. Companies can use Big data tools to improve their online presence

- **Using Big Data Analytics to Boost Customer Acquisition and Retention.**

  Customers are a crucial asset that each company relies on. Without a strong consumer base, no company can be successful.However, even with a strong consumer base, businesses cannot ignore market rivalry. It will be difficult for businesses to succeed if they do not understand what their consumers desire.It will be difficult for businesses to succeed if they do not understand what their consumers desire. It will result in a loss of customers, which will have a negative impact on business growth.Businesses may use big data analytics to detect customer-related trends and patterns. Customer behaviour analysis is the key to a successful business.

- **Using Big Data Analytics to Solve Advertisers Problem and Offer Marketing Insights**

  All company activities are shaped by big data analytics. It allows businesses to meet client expectations. Big data analytics aids in the modification of a company's product range. It guarantees that marketing initiatives are effective.

- **Big Data Analytics as a Driver of Innovations and Product Development**

  Companies may use big data to innovate and revamp their goods.

1.3 Applications of BIG DATA

All of the data must be recorded and processed, which takes a lot of expertise, resources, and time.Data may be creatively and meaningfully used to provide business benefits. There are three sorts of business applications, each with varying degrees of revolutionary potential as shown in Figure 4.

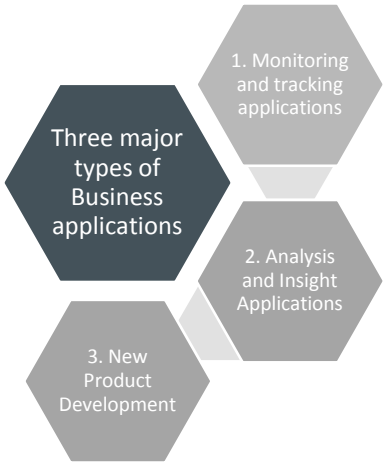


Figure 4 Applications of Big Data

Monitoring and tracking application

These are the first and most fundamental Big Data applications. In practically all industries, they aid in increasing corporate efficiency. The following are a few examples of specialised applications: -

- **Public health monitoring**

  The US government is encouraging all healthcare stakeholders to establish a national platform for interoperability and data sharing standards. This would enable secondary use of health data, which would advance BIG DATA analytics and personalized holistic precision medicine. This would be a broad-based platform like Google flu trends.

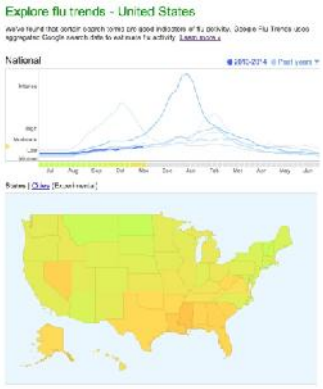


Figure 5 Public health monitoring

- **Consumer Sentiment Monitoring**

  Social media has become more powerful than advertising. Many good companies have moved a bulk of their advertising budgets from traditional media into social media.They have setup Big Data listening platforms, where social media data streams (including tweets, and Facebook posts and blog posts) are filtered and analysed for certain keywords or sentiments, by certain demographics and regions. Actionable information from this analysis is delivered to marketing professionals for appropriate action, especially when the product is new to the market.

Figure 6 Consumer Sentiment Monitoring

- **Asset Tracking**

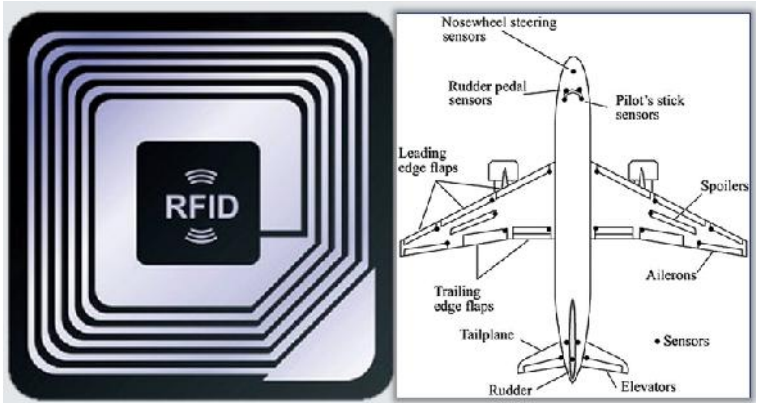


Figure 7 Asset Tracking

The US department of defence is encouraging the industry to devise a tiny RFID chip that could prevent the counterfeiting of electronic parts that end up in avionics or circuit board for other devices. Airplanes are one of the heaviest users of sensors which track every aspect of the performance of every part of the plane. The data can be displayed on the dashboard as well as stored for later detailed analysis. Working with communicating devices, these sensors can produce a torrent of data.Theft by shoppers and employees is a major source of loss of revenue for retailers. All valuable items in the store can be assigned RFID tags, and the gates of the store can be equipped with RF readers. This can help secure the products, and reduce leakage(theft) from the store.

- **Supply chain monitoring**

All containers on ships communicate their status and location using RFID tags. Thus retailers and their suppliers can gain real-time visibility to the inventory throughout the global supply chain. Retailers can know exactly where the items are in the warehouse, and so can bring them into the store at the right time. This is particularly relevant for seasonal items that must be sold on time, or else they will be sold at a discount.With item-level RFID tacks, retailers also gain full visibility of each item and can serve their customers better.

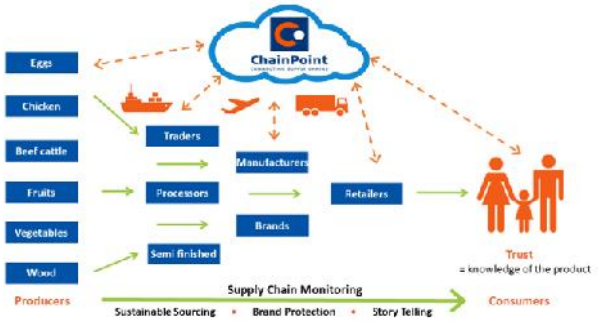


Figure 8 Supply chain monitoring

- **Preventive machine maintenance**

  All machines, including cars and computers, do tend to fail sometimes. This is because one or more or their components may cease to function. As a preventive measure, precious equipment could be equipped with sensors.The continuous stream of data from the sensors could be monitored and analyzed to forecast the status of key components, and thus, monitor the overall machine's health. Preventive maintenance can, thus, reduce the cost of downtime.

Figure 9 Preventive maintenance

Analysis and Insight Applications

These are the next generation of big data apps. They have the ability to improve corporate effectiveness and have transformational potential.Big Data may be organised and analysed to reveal trends and insights that can be utilised to improve business.

- Predictive Policing
- Winning political elections
- Personal Health

Predictive Policing

The notion of predictive policing was created by the Los Angeles Police Department. The LAPD collaborated with UC Berkeley academics to examine its massive database of 13 million crimes spanning 80 years and forecast the likelihood of particular sorts of crimes occurring at specific times and in specific areas.They pinpointed crime hotspots of certain categories, at specific times, and in specific areas.They identified crime hotspots where crimes have happened and were likely to occur in the future.After a basic insight derived from a metaphor of earthquakes and their aftershocks, crime patterns were statistically simulated.The model said that once a crime occurred in a location, it represented a CERTAIN disturbance in harmony, and would thus, lead to a greater likelihood of a similar crime occurring in the local vicinity soon. The model showed for each police beat, the specific neighborhood blocks and specific time slots, where crime was likely to occur.By aligning the police cars patrol schedule in accordance with the models' predictions, the LAPD could reduce crime by 12 percent to 26 percent for different categories of crime.Recently, the SAN Francisco Police department released its own crime for over 2 years, so data analyst could model that data and prevent future crimes.

Figure 10 Predictive policing

Winning political elections

The US president,Barack Obama was the first major political candidate to use big data in a significant way, in the 2008n elections. He is the first big data president. His campaign gathered data about millions of people, including supporters. They invented the mechanism to obtain small campaign contributions from millions of supporters. They

created personal profiles of millions of supporters and what they had done and could do for the campaign. Data was used to determine undecided voters who could be converted to their side. They provided phone numbers of these undecided voters to the volunteers. The results of the calls were recorded in real time using interactive web applications. Obama himself used his twitter account to communicate his message directly with his millions of followers.After the elections, Obama converted his list of tens of millions of supporters to an advocacy machine that would provide the grassroots support for the president initiatives. Since then, almost all campaigns use big data.

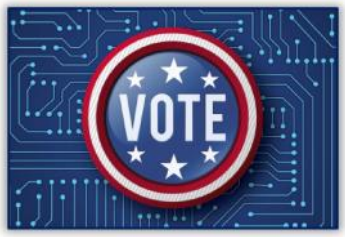


Figure 11 Winning political elections

Senator Bernie sanders used the same big data playbook to build an effective national political machine powered entirely by small donors. Election analyst, Nate silver, created sophistical predictive models using inputs from many political polls and surveys to win pundits to successfully predict winner of the US elections. Nate was however, unsuccessful in predicting Donald trump's rise and ultimate victory and that shows the limits of big data.

Personal health

Medical knowledge and technology is growing by leaps and bounds. IBM's Watson system is a big data analytics engine that ingests and digests all the medical information in the world, and then applies it intelligently to an individual situation.Watson can provide a detailed and accurate medical diagnosis using current symptoms, patient history, medical history and environmental trends, and other parameters. Similar products might be offered as an APP to licensed doctors, and even individuals, to improve productivity and accuracy in health care.

New Product Development

These are completely new notions that did not exist previously. These applications have the ability to disrupt whole sectors and provide organisations with new revenue streams.

- Flexible Auto Insurance
- Location based retail promotion
- Recommendation service

Flexible Auto Insurance

An auto insurance company can use the GPS data from cars to calculate the risk of accidents based on travel patterns. The automobile companies can use the car sensor data to track the performance of a car. Safer drivers can be rewarded and the errant drivers can be penalized.

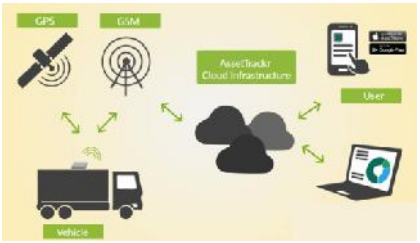


Figure 12 GPS vehicle tracking system

Location based retail promotion

A retailer or a third-party advertiser, can target customers with specific promotions and coupons based on location data obtained through Global positioning system (GPS) the time of day, the presence of stores nearby, and mapping it to the consumer preference data available from social media databases. Advertisements and offers can be delivered through mobile apps, SMS and email. These are examples of mobile apps.

Figure 13 Location based retail promotion

Recommendation service

Ecommerce has been a fast-growing industry in the last couple of decades. A variety of products are sold and shared over the internet. Web users browsing and purchase history on ecommerce sites is utilized to learn about their preference and needs, and to advertise relevant product and pricing offers in real-time. Amazon uses a personalized recommendation engine system to suggest new additional products to consumers based on affinities of various products.

Figure 14 Recommendation Service

Netflix also use a recommendation engine to suggest entertainment options to its users.Big data is valuable across all industries.

These are three major types of data sources of big data. Example (people to people communication, people-machine communications, Machine-machine communications.)Each type has many sources of data. There are three types of applications. They are the monitoring type, the analysis type and new product development.They have an impact on efficiency, effectiveness and even disruption of industries.

1.4 Tools used in BIG DATA

There are number of tools used in BIGDATA. Most popular tools are: -

Apache Hadoop

A large data framework is the Apache Hadoop software library. It enables massive data sets to be processed across clusters of computers in a distributed manner. It's one of the most powerful big data technologies, with the ability to grow from a single server to thousands of computers.

Features

- When utilising an HTTP proxy server, authentication is improved.
- Hadoop Compatible Filesystem effort specification. Extended characteristics for POSIX-style filesystems are supported.

- It has big data technologies and tools that offers robust ecosystem that is well suited to meet the analytical needs of developer.
- It brings Flexibility in Data Processing. It allows for faster data Processing

HPCC

HPCC is a big data tool developed by LexisNexis Risk Solution. It delivers on a single platform, a single architecture and a single programming language for data processing.

Features

- It is one of the Highly efficient big data tools that accomplish big data tasks with far less code.
- It is one of the big data processing tools which offers high redundancy and availability.
- It can be used both for complex data processing on a Thor cluster. Graphical IDE for simplifies development, testing and debugging. It automatically optimizes code for parallel processing
- Provide enhance scalability and performance. ECL code compiles into optimized C++, and it can also extend using C++ libraries

Apache STORM

Storm is a free big data open source computation system. It is one of the best big data tools which offers distributed real-time, fault-tolerant processing system. With real-time computation capabilities.

Features

- It is one of the best tool from big data tools list which is benchmarked as processing one million 100 byte messages per second per node
- It has big data technologies and tools that uses parallel calculations that run across a cluster of machines.
- It will automatically restart in case a node die. The worker will be restarted on another node. Storm guarantees that each unit of data will be processed at least once or exactly once
- Once deployed Storm is surely easiest tool for Bigdata analysis

Qubole

Qubole Data is Autonomous Big data management platform. It is a big data open-source tool which is self-managed, self-optimizing and allows the data team to focus on business outcomes.

Features

- **Features:**
- Single Platform for every use case
- It is an Open-source big data software having Engines, optimized for the Cloud.
- Comprehensive Security, Governance, and Compliance
- Provides actionable Alerts, Insights, and Recommendations to optimize reliability, performance, and costs.
- Automatically enacts policies to avoid performing repetitive manual actions

Apache Cassandra

The Apache Cassandra database is widely used today to provide an effective management of large amounts of data.

Features

- Support for replicating across multiple data centers by providing lower latency for users
- Data is automatically replicated to multiple nodes for fault-tolerance
- It one of the best big data tools which is most suitable for applications that can't afford to lose data, even when an entire data center is down
- Cassandra offers support contracts and services are available from third parties
-

Statwing

Statwing is an easy-to-use statistical tool. It was built by and for big data analysts. Its modern interface chooses statistical tests automatically.

Features

- It is a big data software that can explore any data in seconds. Statwing helps to clean data, explore relationships, and create charts in minutes
- It allows creating histograms, scatterplots, heatmaps, and bar charts that export to Excel or PowerPoint. It also translates results into plain English, so analysts unfamiliar with statistical analysis

CouchDB

CouchDB stores data in JSON documents that can be accessed web or query using JavaScript. It offers distributed scaling with fault-tolerant storage. It allows accessing data by defining the Couch Replication Protocol.

Features

- CouchDB is a single-node database that works like any other database
- It is one of the big data processing tools that allows running a single logical database server on any number of servers.
- It makes use of the ubiquitous HTTP protocol and JSON data format. Easy replication of a database across multiple server instances. Easy interface for document insertion, updates, retrieval and deletion
- JSON-based document format can be translatable across different languages

Pentaho

Pentaho provides big data tools to extract, prepare and blend data. It offers visualizations and analytics that change the way to run any business. This Big data tool allows turning big data into big insights.

Features:

- Data access and integration for effective data visualization. It is a big data software that empowers users to architect big data at the source and stream them for accurate analytics.

- Seamlessly switch or combine data processing with in-cluster execution to get maximum processing. Allow checking data with easy access to analytics, including charts, visualizations, and reporting
- Supports wide spectrum of big data sources by offering unique capabilities

Apache Flink

Apache Flink is one of the best open source data analytics tools for stream processing big data. It is distributed, high-performing, always-available, and accurate data streaming applications.

Features:

- Provides results that are accurate, even for out-of-order or late-arriving data
- It is stateful and fault-tolerant and can recover from failures.
- It is a big data analytics software which can perform at a large scale, running on thousands of nodes
- Has good throughput and latency characteristics
- This big data tool supports stream processing and windowing with event time semantics. It supports flexible windowing based on time, count, or sessions to data-driven windows
- It supports a wide range of connectors to third-party systems for data sources and sinks

Cloudera

Cloudera is the fastest, easiest and highly secure modern big data platform. It allows anyone to get any data across any environment within single, scalable platform.

Features:

- High-performance big data analytics software
- It offers provision for multi-cloud
- Deploy and manage Cloudera Enterprise across AWS, Microsoft Azure and Google Cloud Platform. Spin up and terminate clusters, and only pay for what is needed when need it
- Developing and training data models
- Reporting, exploring, and self-servicing business intelligence
- Delivering real-time insights for monitoring and detection
- Conducting accurate model scoring and serving

Open Refine

OpenRefine is a powerful big data tool. It is a big data analytics software that helps to work with messy data, cleaning it and transforming it from one format into another. It also allows extending it with web services and external data.

Features:

- OpenRefine tool help you explore large data sets with ease. It can be used to link and extend your dataset with various webservices. Import data in various formats.
- Explore datasets in a matter of seconds
- Apply basic and advanced cell transformations
- Allows to deal with cells that contain multiple values

- Create instantaneous links between datasets. Use named-entity extraction on text fields to automatically identify topics. Perform advanced data operations with the help of Refine Expression Language

RapidMiner

RapidMiner is one of the best open-source data analytics tools. It is used for data prep, machine learning, and model deployment. It offers a suite of products to build new data mining processes and setup predictive analysis.

Features

- Allow multiple data management methods
- GUI or batch processing
- Integrates with in-house databases
- Interactive, shareable dashboards
- Big Data predictive analytics
- Remote analysis processing
- Data filtering, merging, joining and aggregating
- Build, train and validate predictive models
- Store streaming data to numerous databases
- Reports and triggered notifications

Data cleaner

Data Cleaner is a data quality analysis application and a solution platform. It has strong data profiling engine. It is extensible and thereby adds data cleansing, transformations, matching, and merging.

Feature:

- Interactive and explorative data profiling
- Fuzzy duplicate record detection.
- Data transformation and standardization
- Data validation and reporting
- Use of reference data to cleanse data
- Master the data ingestion pipeline in Hadoop data lake. Ensure that rules about the data are correct before user spends thier time on the processing. Find the outliers and other devilish details to either exclude or fix the incorrect data

Kaggle

Kaggle is the world's largest big data community. It helps organizations and researchers to post their data & statistics. It is the best place to analyze data seamlessly.

Features:

- The best place to discover and seamlessly analyze open data
- Search box to find open datasets.

- Contribute to the open data movement and connect with other data enthusiasts

Apache Hive

Hive is an open-source big data software tool. It allows programmers analyze large data sets on Hadoop. It helps with querying and managing large datasets real fast.

Features:

- It Supports SQL like query language for interaction and Data modeling
- It compiles language with two main tasks map, and reducer.
- It allows defining these tasks using Java or Python
- Hive designed for managing and querying only structured data
- Hive's SQL-inspired language separates the user from the complexity of Map Reduce programming
- It offers Java Database Connectivity (JDBC) interface

1.5 Challenges in BIG DATA

Lack of proper understanding of Big Data

Companies fail in their Big Data initiatives due to insufficient understanding. Employees may not know what data is, its storage, processing, importance, and sources. Data professionals may know what is going on, but others may not have a clear picture.For example, if employees do not understand the importance of data storage, they might not keep the backup of sensitive data. They might not use databases properly for storage. As a result, when this important data is required, it cannot be retrieved easily.
Big Data workshops and seminars must be held at companies for everyone. Basic training programs must be arranged for all the employees who are handling data regularly and are a part of the Big Data projects. A basic understanding of data concepts must be inculcated by all levels of the organization.

Data growth issues

One of the most pressing challenges of Big Data is storing all these huge sets of data properly. The amount of data being stored in data centers and databases of companies is increasing rapidly. As these data sets grow exponentially with time, it gets extremely difficult to handle.Most of the data is unstructured and comes from documents, videos, audios, text files and other sources. This means that you cannot find them in databases.

Solution

In order to handle these large data sets, companies are opting for modern techniques, such as compression, tiering, and deduplication. Compression is used for reducing the number of bits in the data, thus reducing its overall size. Deduplication is the process of removing duplicate and unwanted data from a data set. Data tiering allows companies to store data in different storage tiers. It ensures that the data is residing in the most appropriate storage space. Data tiers can be public cloud, private cloud, and flash storage, depending on the data size and importance. Companies are also opting for Big Data tools, such as Hadoop, NoSQL and other technologies. This leads us to the third Big Data problem.

Confusion while Big Data tool selection

- Companies often get confused while selecting the best tool for Big Data analysis and storage. Is HBase or Cassandra the best technology for data storage? Is Hadoop MapReduce good enough or will Spark be a better option for data analytics and storage?These questions bother companies and sometimes they are unable to find the

answers. They end up making poor decisions and selecting an inappropriate technology. As a result, money, time, efforts and work hours are wasted.

Solution

The best way to go about it is to seek professional help. You can either hire experienced professionals who know much more about these tools. Another way is to go for Big Data consulting. Here, consultants will give a recommendation of the best tools, based on your company's scenario. Based on their advice, you can work out a strategy and then select the best tool for you.

Lack of data professionals

- To run these modern technologies and Big Data tools, companies need skilled data professionals. These professionals will include data scientists, data analysts and data engineers who are experienced in working with the tools and making sense out of huge data sets.Companies face a problem of lack of Big Data professionals. This is because data handling tools have evolved rapidly, but in most cases, the professionals have not. Actionable steps need to be taken in order to bridge this gap.

Solution

Companies are investing more money in the recruitment of skilled professionals. They also have to offer training programs to the existing staff to get the most out of them.Another important step taken by organizations is the purchase of data analytics solutions that are powered by artificial intelligence/machine learning. These tools can be run by professionals who are not data science experts but have basic knowledge. This step helps companies to save a lot of money for recruitment.

Securing data

- Securing these huge sets of data is one of the daunting challenges of Big Data. Often companies are so busy in understanding, storing and analyzing their data sets that they push data security for later stages. But, this is not a smart move as unprotected data repositories can become breeding grounds for malicious hackers.
- Companies can lose up to **$3.7 million** for a stolen record or a data breach.
- Solution
- Companies are recruiting more cybersecurity professionals to protect their data. Other steps taken for securing data include:
- Data encryption
- Data segregation
- Identity and access control
- Implementation of endpoint security
- Real-time security monitoring

Integrating data from a variety of sources

- Data in an organization comes from a variety of sources, such as social media pages, ERP applications, customer logs, financial reports, e-mails, presentations and reports created by employees. Combining all this data to prepare reports is a challenging task.This is an area often neglected by firms. But, data integration is crucial for analysis, reporting and business intelligence, so it has to be perfect.

Solution

Companies have to solve their data integration problems by purchasing the right tools. Some of the best data integration tools are mentioned below:

- Talend Data Integration
- Centerprise Data Integrator
- ArcESB

- IBM InfoSphere
- Xplenty
- Informatica PowerCenter
- CloverDX
- Microsoft SQL
- QlikView
- Oracle Data Service Integrator

In order to put Big Data to the best use, companies have to start doing things differently. This means hiring better staff, changing the management, reviewing existing business policies and the technologies being used. To enhance decision making, they can hire a Chief Data Officer – a step that is taken by many of the fortune 500 companies.

Summary

- Big data refers to massive, difficult-to-manage data quantities – both organised and unstructured – that inundate enterprises on a daily basis. Big data may be evaluated for insights that help people make better judgments and feel more confident about making key business decisions.
- These are the most basic and basic Big Data applications. They assist in enhancing company efficiency in almost every industry.
- These are the big data apps of the future. They have the potential to alter businesses and boost corporate effectiveness. Big data may be organised and analysed to uncover patterns and insights that can be used to boost corporate performance.
- These are brand-new concepts that didn't exist before. These applications have the potential to disrupt whole industries and generate new income streams for businesses.
- Apache Hadoop is a set of open-source software tools for solving issues involving large volumes of data and processing utilising a network of many computers. It uses the MapReduce programming concept to create a software framework for distributed storage and processing of massive data.
- Apache Cassandra is a distributed, wide-column store, NoSQL database management system that is designed to handle massive volumes of data across many commodity servers while maintaining high availability and avoiding single points of failure.
- Cloudera, Inc. is a Santa Clara, California-based start-up that offers a subscription-based enterprise data cloud. Cloudera's platform, which is based on open-source technology, leverages analytics and machine learning to extract insights from data through a secure connection.
- RapidMiner is a data science software platform built by the same-named firm that offers a unified environment for data preparation, machine learning, deep learning, text mining, and predictive analytics.
- Kaggle, a Google LLC subsidiary, is an online community of data scientists and machine learning experts.
- LexisNexis Risk Solutions created HPCC, often known as DAS, an open source data-intensive computing system platform. The HPCC platform is based on a software architecture that runs on commodity computing clusters and provides high-performance, data-parallel processing for big data applications.

Keywords

Big Data: Big data refers to massive, difficult-to-manage data quantities – both organised and unstructured – that inundate enterprises on a daily basis. But it's not simply the type or quantity of data that matters; it's also what businesses do with it. Big data may be evaluated for insights that help people make better judgments and feel more confident about making key business decisions.

Volume: Transactions, smart (IoT) devices, industrial equipment, videos, photos, audio, social media, and other sources are all used to collect data. Previously, keeping all of that data would have been too expensive; now, cheaper storage options such as data lakes, Hadoop, and the cloud have alleviated the strain.

Velocity:Data floods into organisations at an unprecedented rate as the Internet of Things grows, and it must be handled quickly. The need to cope with these floods of data in near-real time is being driven by RFID tags, sensors, and smart metres.

Variety: From organised, quantitative data in traditional databases to unstructured text documents, emails, movies, audios, stock ticker data, and financial transactions, data comes in a variety of formats.

Variability: Data flows are unpredictable, changing often and altering substantially, in addition to rising velocities and variety of data. It's difficult, but companies must recognise when something is hot on social media and how to manage high data loads on a daily, seasonal, and event-triggered basis.

Veracity:The quality of data is referred to as veracity. Information's tough to link, match, cleanse, and convert data across systems since it originates from so many diverse places. Relationships, hierarchies, and numerous data links must all be connected and correlated by businesses. If they don't, their data will rapidly become out of hand.

Self Assessment

Q1: What are the fundamental elements of BIG DATA?

A. HDFS
B. YARN
C. MapReduce
D. All of these

Q2: What distinguishes BIG DATA Analytics from other types of analytics?

A. Open-Source
B. Scalability
C. Data Recovery
D. All of these

Q3: What are the Big Data V's?

A. Volume
B. Veracity
C. Both a and b
D. Vivid

Q4: Please identify the right statement.

A. Hadoop is an excellent platform for extracting and analyzing tiny amounts of data.
B. Hadoop uses HDFS to store data and enables data compression and decompression.

C. To solve graph and machine learning problems, the giraph framework is less useful than a MapReduce framework.

D. None of the mentioned

Q5: On which of the following platforms, Hadoop is available.

A. Bare metal

B. Cross-Platform

C. Unix-Like

D. None of the mentioned

Q6: The Hadoop list includes the HBase database, the Apache Mahout _____ System, and matrix operations.

A. Pattern recognition

B. HPCC

C. Machine Learning

D. SPSS

Q7: The element of MapReduce is in charge of processing one or more data chunks and providing output results.

A. MapTask

B. Mapper

C. Task execution

D. All of the mentioned

Q8: Although the Hadoop framework is implemented in Java, MapReduce applications need not be written in _____

A. Java

B. C

C. C#

D. None of the mentioned

Q9: Input key/value pairs are mapped to a collection of intermediate key/value pairs using _____.

A. Mapper

B. Reducer

C. Both

D. None of the mentioned

Q10: The number of maps is usually driven by the total size of _____

A. inputs

B. outputs

C. tasks

D. None of the mentioned

Q11: The _____ software library is a big data framework. It allows distributed processing of large data sets across clusters of computers.

A. Apple Programming
B. R Programming
C. Apache Hadoop
D. All of above

Q12: Which big data tool was developed by LexisNexis Risk Solution?

A. SPCC System
B. HPCC System
C. TOCC System
D. None of above

Q13: Which big data tools offers distributed real-time, fault-tolerant processing system with real-time computation capabilities.

A. Storm
B. HPCC
C. Qubole
D. Cassandra

Q14: Which statement regarding Apache Cassandra is correct?

A. It is free and open-source tool.
B. It is widely used today to provide an effective management of large amounts of data.
C. It is distributed
D. All of the above

Q15: _____ stores data in JSON documents that can be accessed web or query using JavaScript

A. CouchDB
B. Storm
C. Hive
D. None of above

Answers for Self Assessment

1.	D	2.	D	3.	C	4.	B	5.	B
6.	C	7.	C	8.	A	9.	A	10.	A
11.	C	12.	B	13.	A	14.	D	15.	A

Review Questions

1. Explain five effective characteristics of BIG DATA.
2. Write down applications of BIG DATA.

3. How BIG DATA Vs. are are classified? Explain in detail.
4. Write down challenges of BIG DATA.
5. Explain the difference between Volume, Veracity and Velocity.
6. Write down the tools used in BIG DATA.

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think. Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley. White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 02: Foundations for Big Data

CONTENTS

Objectives

- differentiate between file system (FS) and distributed file system (DFS)
- understand scalable computing over the internet.
- understand programming models for Big Data.

Introduction

The first storage mechanism used by computers to store data was punch cards. Each group of related punch cards (Punch cards related to same program) used to be stored into a file; and files were stored in file cabinets. This is very similar to what we do nowadays to archive papers in government intuitions who still use paper work on daily basis. This is where the word "File System" (FS) comes from. The computer systems evolved; but the concept remains the same.

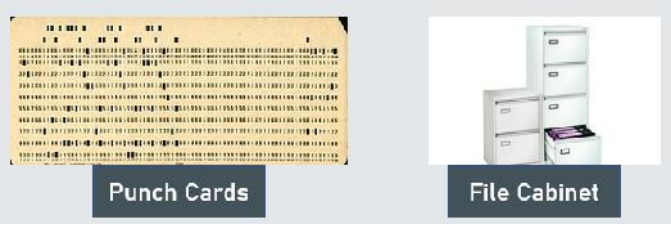


Figure 1 Storage Mechanism

2.1 What is File System?

Instead of storing information on punch cards; we can now store information / data in a digital format on a digital storage device such as hard disk, flash drive…etc. Related data are still categorized as files; related groups of files are stored in folders. Each file has a name, extension and icon. The file name gives an indication about the content it has while

file extension indicates the type of information stored in that file. for example; EXE extension refers to executable files, TXT refers to text files…etc.File management system is used by the operating system to access the files and folders stored in a computer or any external storage devices.

Figure 2 Digital Storage

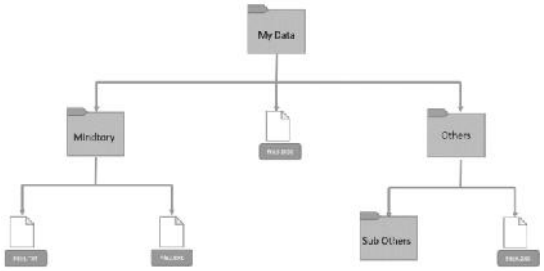
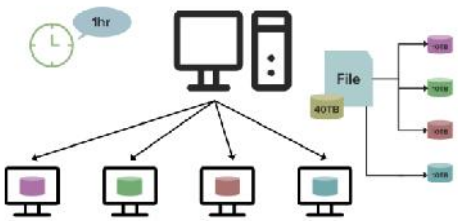


Figure 3 Example to File System

2.2 What is Distributed File System?

In Big Data, we deal with multiple clusters (computers) often. One of the main advantages of Big Data which is that it goes beyond the capabilities of one single super powerful server with extremely high computing power. The whole idea of Big Data is to distribute data across multiple clusters and to make use of computing power of each cluster (node) to process information.Distributed file system is a system that can handle accessing data across multiple clusters (nodes). In the next section we will learn more about how it works?



DFS has two components

Location Transparency: Location Transparency achieves through the namespace component.

Redundancy: Redundancy is done through a file replication component.

Features of DFS

Transparency

- ***Structure transparency*:** There is no need for the client to know about the number or locations of file servers and the storage devices. Multiple file servers should be provided for performance, adaptability, and dependability.
- ***Access transparency*:** Both local and remote files should be accessible in the same manner. The file system should be automatically located on the accessed file and send it to the client's side.

- ***Naming transparency*:** There should not be any hint in the name of the file to the location of the file. Once a name is given to the file, it should not be changed during transferring from one node to another.
- ***Replication transparency*:** If a file is copied on multiple nodes, both the copies of the file and their locations should be hidden from one node to another.

User mobility: It will automatically bring the user's home directory to the node where the user logs in.

Performance: Performance is based on the average amount of time needed to convince the client requests. This time covers the CPU time + time taken to access secondary storage + network access time. It is advisable that the performance of the Distributed File System be similar to that of a centralized file system.

Simplicity and ease of use: The user interface of a file system should be simple and the number of commands in the file should be small.

High availability: A Distributed File System should be able to continue in case of any partial failures like a link failure, a node failure, or a storage drive crash.
A high authentic and adaptable distributed file system should have different and independent file servers for controlling different and independent storage devices.

How Distributed file system (DFS) works?

Distributed file system works as follows:

- **Distribution:** Distribute blocks of data sets across multiple nodes. Each node has its own computing power; which gives the ability of DFS to parallel processing data blocks.
- **Replication**: Distributed file system will also replicate data blocks on different clusters by copy the same pieces of information into multiple clusters on different racks. This will help to achieve the following:
    - ***Fault Tolerance***: recover data block in case of cluster failure or Rack failure.Data replication is a good way to achieve fault tolerance and high concurrency; but it's very hard to maintain frequent changes. Assume that someone changed a data block on one cluster; these changes need to be updated on all data replica of this block.

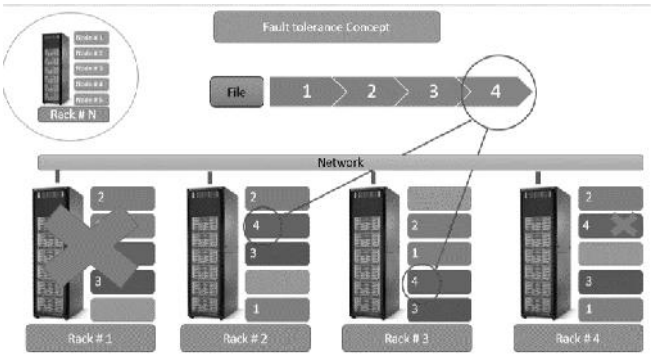


Figure 4 Fault Tolerance Concept

Data replication is a good way to achieve fault tolerance and high concurrency; but it's very hard to maintain frequent changes. Assume that someone changed a data block on one cluster; these changes need to be updated on all data replica of this block.

- ***High Concurrency***: avail same piece of data to be processed by multiple clients at the same time. It is done using the computation power of each node to parallel process data blocks.

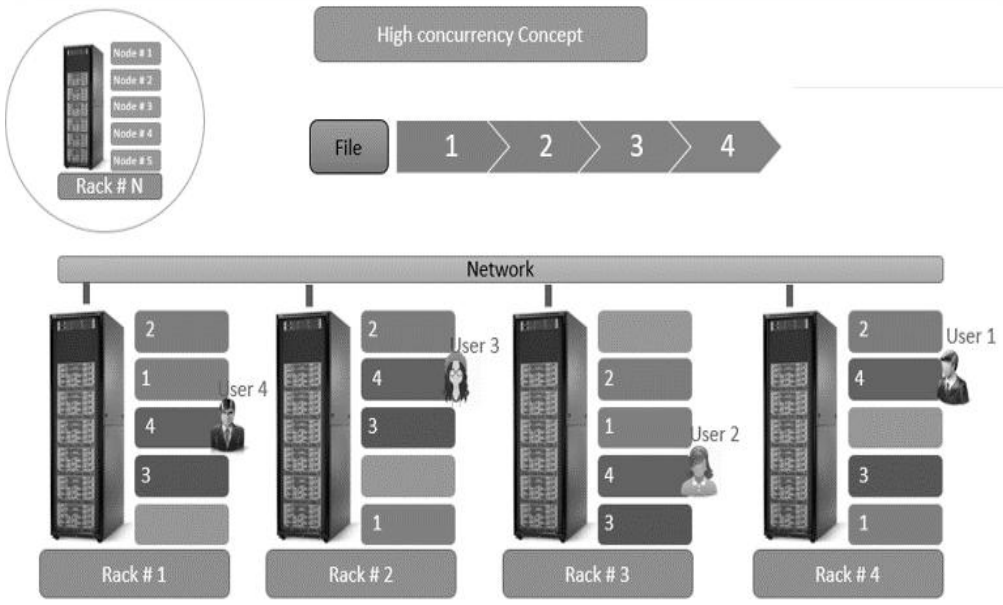


Figure 5 High Tolerance Concept

What are the Advantages of Distributed File System (DFS)?

- **Scalability**: You can scale up your infrastructure by adding more racks or clusters to your system.
- **Fault Tolerance**: Data replication will help to achieve fault tolerance in the following cases:
    - Cluster is down
    - Rack is down
    - Rack is disconnected from the network.
    - Job failure or restart.
- **High Concurrency**: utilize the compute power of each node to handle multiple client requests (in a parallel way) at the same time. The following figure illustrates the main concept of high concurrency and how it can be achieved by data replication on multiple clusters.
- DFS allows multiple users to access or store the data.
- It allows the data to be share remotely.
- It improved the availability of file, access time and network efficiency.
- Improved the capacity to change the size of the data and also improves the ability to exchange the data.
- Distributed File System provides transparency of data even if server or disk fails.

What are the disadvantages of Distributed File System (DFS)?

- In Distributed File System nodes and connections needs to be secured therefore we can say that security is at stake.
- There is a possibility of loss of messages and data in the network while movement from one node to another.
- Database connection in case of Distributed File System is complicated.
- Also handling of the database is not easy in Distributed File System as compared to a single user system.
- There are chances that overloading will take place if all nodes try to send data at once.

2.3 Scalable Computing Over the Internet

What is Scalability?

- With Cloud hosting, it is easy to grow and shrink the number and size of servers based on the need.This is done by either increasing or decreasing the resources in the cloud.This ability to alter plans due to fluctuation in business size and needs is a superb benefit of cloud computing especially when experiencing a sudden growth in demand.

Scalable Computing Over the Internet

- *The Age of Internet Computing*
- *High-Performance Computing*
- *High-Throughput Computing*
- *Three New Computing Paradigms*
- *Computing Paradigm Distinctions*
- *Distributed System Families*
- *Degrees of Parallelism*

The Age of Internet Computing

Billions of people use the Internet every day. As a result, supercomputer sites and large data centers must provide high-performance computing services to huge numbers of Internet users concurrently.Because of this high demand, the Linpack Benchmark for high-performance computing (HPC) applications is no longer optimal for measuring system performance.The emergence of computing clouds instead demands high-throughput computing (HTC) systems built with parallel and distributed computing technologies [5,6,19,25]. We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks. The purpose is to advance network-based computing and web services with the emerging new technologies.

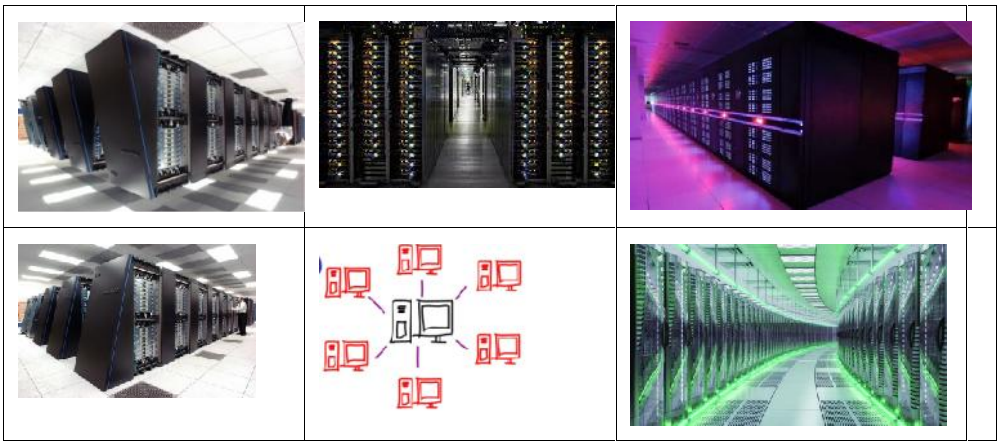


Figure 6 The Age of Internet Computing

The platform Evolution

We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks. The purpose is to advance network-based computing and web services with the emerging new technologies.On the HPC side, supercomputers (massively parallel processors or MPPs) are gradually replaced by clusters of cooperative computers out of a desire to share computing resources. The cluster is often a collection of homogeneous compute nodes that are physically connected in close range to one another.

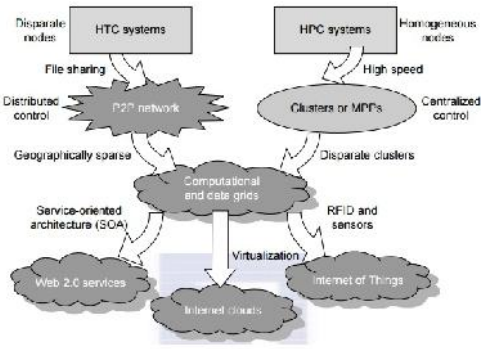


Figure 7 Platform Evolution

On the HTC side, peer-to-peer (P2P) networks are formed for distributed file sharing and content delivery applications.

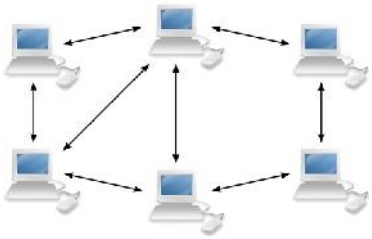


Figure 8 Peer to Peer Network

A P2P system is built over many client machines. Peer machines are globally distributed in nature. P2P, cloud computing, and web service platforms are more focused on HTC applications than on HPC applications.Clustering and P2P technologies lead to the development of computational grids or data grids.

High-Performance Computing

- For many years, HPC systems emphasize the raw speed performance. The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010. This improvement was driven mainly by the demands from scientific, engineering, and manufacturing communities.For example, the Top 500 most powerful computer systems in the world are measured by floating-point speed in Linpack benchmark results. However, the number of supercomputer users is limited to less than 10% of all computer users. Today, the majority of computer users are using desktop computers or large servers when they conduct Internet searches and market-driven computing tasks.

High Throughput Computing

- The development of market-oriented high-end computing systems is undergoing a strategic change from an HPC paradigm to an HTC paradigm. This HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously.The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time. HTC technology needs to not only improve in terms of batch processing speed, but also address the acute problems of cost, energy savings, security, and reliability at many data and enterprise computing centers.

Three New Computing Paradigms

- Advances in virtualization make it possible to see the growth of Internet clouds as a new computing paradigm. The maturity of radio-frequency identification (RFID), Global Positioning System (GPS), and sensor technologies has triggered the development of the Internet of Things (IoT).

Figure 9 Three New Computing Paradigms

Computing Paradigm Distinctions

The high-technology community has argued for many years about the precise definitions of centralized computing, parallel computing, distributed computing, and cloud computing. Distributed computing is the opposite of centralized computing.The field of parallel computing overlaps with distributed computing to a great extent, and cloud computing overlaps with distributed, centralized, and parallel computing.Centralized computing This is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS. Many data centers and supercomputers are centralized systems, but they are used in parallel, distributed, and cloud computing applications. Parallel computing In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Some authors refer to this discipline as parallel processing. Interprocessor communication is accomplished through shared memory or via message passing. A computer system capable of parallel computing is commonly known as a parallel computer [28]. Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often referred to as parallel programming

Distributed System Families

Since the mid-1990s, technologies for building P2P networks and networks of clusters have been consolidated into many national projects designed to establish wide area computing infrastructures, known as computational grids or data grids.Recently, we have witnessed a surge in interest in exploring Internet cloud resources for data-intensive applications. Internet clouds are the result of moving desktop computing to service-oriented computing using server clusters and huge databases at data centers. Grids and clouds are disparity systems that place great emphasis on resource sharing in hardware, software, and data sets.

Degrees of Parallelism

- Fifty years ago, when hardware was bulky and expensive, most computers were designed in a bit-serial fashion.In this scenario, bit-level parallelism (BLP) converts bit-serial processing to word-level processing gradually.Over the years, users graduated from 4-bit microprocessors to 8-,16-, 32-, and 64-bit CPUs. This led us to the next wave of improvement, known as instruction-level parallelism (ILP), in which the processor executes multiple instructions simultaneously rather than only one instruction at a time. For the past 30 years, we have practiced ILP through pipelining, superscalar computing, VLIW (very long instruction word) architectures, and multithreading.ILP requires branch prediction, dynamic scheduling, speculation, and compiler support to work efficiently

2.4 Popular models for big data are

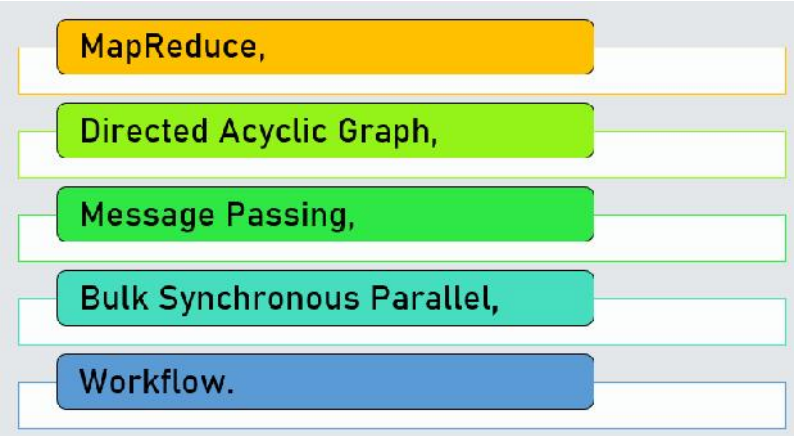


Figure 10 Popular models for big data

What is a MapReduce?

- Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job. Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The Algorithm

- Generally, MapReduce paradigm is based on sending the computer to where the data resides. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
    - **Map stage** − The map or mapper's job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
    - **Reduce stage** − This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

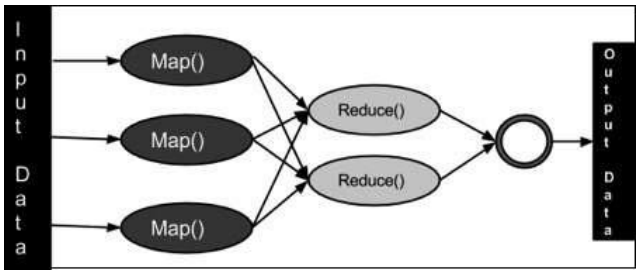


Figure 11 MapReduce

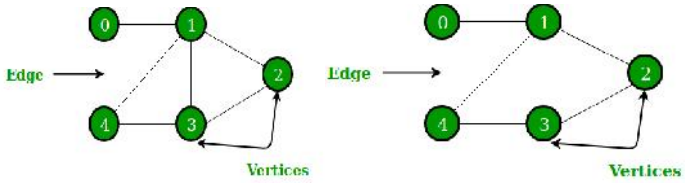
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic. After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Advantages of MapReduce

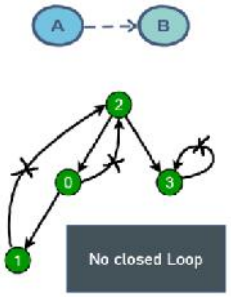
- It is easy to scale data processing over multiple computing nodes.
- Under the MapReduce model, the data processing primitives are called mappers and reducers.
- Decomposing a data processing application into mappers and *reducers* is sometimes nontrivial.
- But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster are merely a configuration change.
- This simple scalability is what has attracted many programmers to use the MapReduce model.

Directed Acyclic Graph

In computer science and mathematics, a directed acyclic graph (DAG) refers to a directed graph which has no directed cycles.In graph theory, a graph refers to a set of vertices which are connected by lines called edges.



In a directed graph or a digraph, each edge is associated with a direction from a start vertex to an end vertex.If we traverse along the direction of the edges and we find that no closed loops are formed along any path, we say that there are no directed cycles. The graph formed is a directed acyclic graph.A DAG is always topologically ordered, i.e., for each edge in the graph, the start vertex of the edge occurs earlier in the sequence than the ending vertex of the edge.



Topological sorting for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge u v, vertex u comes before v in the ordering. Topological Sorting for a graph is not possible if the graph is not a DAG.For example, a topological sorting of the following graph is "5 4 2 3 1 0". There can be more than one topological sorting for a graph. For example, another topological sorting of the following graph is "4 5 2 3 1 0". The first vertex in topological sorting is always a vertex with in-degree as 0 (a vertex with no incoming edges).

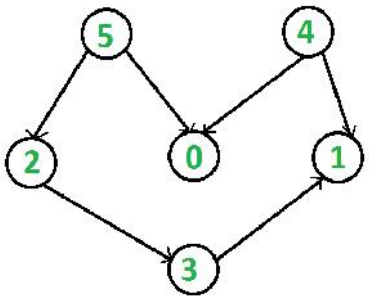


Figure 12 Topological Sorting

Application Areas

Some of the main application areas of DAG are −

- Routing in computer networks
- Job scheduling
- Data processing
- Genealogy
- Citation graphs

Message Passing

Process communication is the mechanism provided by the operating system that allows processes to communicate with each other. This communication could involve a process letting another process know that some event has occurred or transferring of data from one process to another. One of the models of process communication is the message passing model.Message passing model allows multiple processes to read and write data to the message queue without being connected to each other. Messages are stored on the queue until their recipient retrieves them. Message queues are quite useful for inter-process communication and are used by most operating systems.

In the above diagram, both the processes P1 and P2 can access the message queue and store and retrieve data.The message passing model is much easier to implement than the shared memory model.It is easier to build parallel hardware using message passing model as it is quite tolerant of higher communication latencies.

Disadvantages of Message Passing

The message passing model has slower communication than the shared memory model because the connection setup takes time.

Bulk Synchronous Parallel

Although widespread, the MapReduce model is not without its drawbacks. When we talk about the model being applied in the context of Hadoop, for example, all of the cluster steps and mounting of the final mass with the results is done through files on the file system of Hadoop, HDFS, which generates an overhead in performance when it has to perform the same processing in a iterative manner.Another problem is that for graph algorithms such as DFS, BFS or Pert, MapReduce model is not satisfactory. For these scenarios, there is the BSP.In the BSP algorithm, we have the concept of super steps.

A super step consists of a unit of generic programming, which through a global communication component, makes thousands of parallel processing on a mass of data and sends it to a "meeting" called synchronization barrier.

At this point, the data are grouped, and passed on to the next superstep chain.

In this model, it is simpler to construct iterative workloads, since the same logic can be re-executed in a flow of super steps. Another advantage pointed out by proponents of this model is that it has a simpler learning curve for developers coming from the procedural world.

And so, we conclude another part of our series on Big Data. To date, these are the main models used by the Big Data platforms. As a technology booming, it is natural that in the future we could have more models emerging and gaining their adoption shares.

2.5 Five Reasons You Need a Step-by-Step Approach to Workflow Orchestration for Big Data

Is your organization struggling to keep up with the demands of Big Data and under pressure to prove quick results? If so, you're not alone. According to analysts, up to 60% of Big Data projects are failing because they can't scale at the enterprise level. Fortunately, taking a step-by-step approach to application workflow orchestration can help you succeed. It begins with assessing the various technologies for supporting multiple Big Data projects that relate to these four steps:

- Ingesting data

- Storing the data
- Processing it
- Making data available for analytics

The approach also requires having a reliable application workflow orchestration tool that simplifies the complexity of Big Data workflows, avoids automation silos, connects processes, and manages workflows from a single point. This allows you end to end automation, integration and orchestration of your Big Data processes, ensuring that everything is running successfully, meeting all SLAs, and delivering insights to business users on time.Cobbling together disparate automation and orchestration tools that don't scale, may cause delays and put the entire project at risk. Here are some of the benefits of beginning your Big Data project with application workflow orchestration in mind and using a tool that supports these steps:

Improve quality, speed, and time to market

Many Big Data projects drag on or fail. If developers don't have the tools to properly scale their efforts, they may either write numerous, hard-to-manage scripts or rely on limited functionality tools for scheduling. Their tools may not integrate well with other processes, such as file transfers. With a workload orchestration solution, you can implement Big Data projects quickly to help retain your customer base and maintain a competitive edge.

Reduce complexity in all environments – on premises, hybrid, and multi-cloud

An orchestration tool that can automate, schedule, and manage processes successfully across the different components in a Big Data project reduces this complexity. It can manage the main steps of data ingestion, storing the data, processing the data, and finally the whole analytics part. It should also provide a holistic view of the different components and technologies they use to orchestrate those workflows.A Big Data workflow usually consists of various steps with multiple technologies and many moving parts. You need to simplify workflows to deliver big data project successfully on time, especially in the cloud, which is the platform of choice for most Big Data projects. The cloud, however, adds to the complexity, so your orchestration solution needs to be platform agnostic, supporting both on-premises and multi-cloud environments.

Ensure scalability and reduce risk

As I mentioned earlier, Big Data projects must be able to scale, especially when you start moving from the pilot phase to production. Processes for developing and deploying Big Data jobs need to be automated and repeatable. Once the pilot runs successfully, other parts of the business will look into taking advantage of Big Data projects as well. Your workload orchestration solution should make it easy scale and support the growing business demands.

Achieve better Integration

Big Data automation open-source solutions have generally limited capabilities and lack essential management features. More than that, they tend to be limited to a specific environment (ie Hadoop) but keep in mind that Big Data is not an island. It often needs to integrate with other parts of the business. So, your Big Data projects should be connected with upstream and downstream applications, platforms and data sources (ie ERP systems, EDW etc) our big data orchestration solution should provide this capability.

Improve reliability

it's important to run Big Data workflows successfully to minimize service interruptions. Using a patchwork of tools and processes makes it hard to identify issues and understand root cause, putting SLAs at risk. If you can manage your entire Big Data workflow from A to Z, then if something goes wrong in the process, you'll see it immediately and know where it happened and what happened. Using the same solution orchestrating your entire processes and managing them from one single plane of glass, simplifies managing your services and assuring they run successfully.

Looking ahead

Taking a step-by-step approach to application workflow orchestration simplifies the complexity of your Big Data workflows. It avoids automation silos and helps assure you meet SLAs and deliver insights to business users on time. Discover how Control-M provides all

ofthe capabilities to enable your organization to follow this approach and how it easily integrates with your existing technologies to support Big Data projects.

Summary

- A file system is a programme that controls how and where data is saved, retrieved, and managed on a storage disc, usually a hard disc drive (HDD). It's a logical disc component that maintains a disk's internal activities as they relate to a computer while remaining invisible to the user.
- A distributed file system (DFS) or network file system is a type of file system that allows many hosts to share files over a computer network. Multiple users on multiple machines can share data and storage resources as a result of this.
- The distinction between local and remote access techniques should be indistinguishable.
- Users who have access to similar communication services at multiple locations are said to be mobile. For example, a user can use a smartphone and access his email account from any computer to check or compose emails. The travel of a communication device with or without a user is referred to as device portability.
- Big data refers to massive, difficult-to-manage data quantities – both organised and unstructured – that inundate enterprises on a daily basis. Big data may be evaluated for insights that help people make better judgments and feel more confident about making key business decisions.
- These are the most basic and basic Big Data applications. They assist in enhancing company efficiency in almost every industry.
- These are the big data apps of the future. They have the potential to alter businesses and boost corporate effectiveness. Big data may be organised and analysed to uncover patterns and insights that can be used to boost corporate performance.
- The process of replicating a double-stranded DNA molecule into two identical DNA molecules is known as DNA replication. Because every time a cell splits, the two new daughter cells must have the same genetic information, or DNA, as the parent cell, replication is required.
- The capacity of a system to increase or decrease in performance and cost in response to changes in application and system processing demands is known as scalability. When considering hardware and software, businesses that are rapidly expanding should pay special attention to scalability.
- In a Hadoop cluster, MapReduce is a programming paradigm that permits tremendous scalability across hundreds or thousands of computers. MapReduce, as the processing component, lies at the heart of Apache Hadoop. The reduction job is always carried out after the map job, as the term MapReduce implies.

Keywords

MapReduce: MapReduce is a framework that allows us to create applications that reliably process enormous volumes of data in parallel on vast clusters of commodity hardware.

Map Stage: The map's or mapper's job is to process the data that is given to them. In most cases, the input data is stored in the Hadoop file system as a file or directory (HDFS). Line by line, the input file is supplied to the mapper function. The mapper divides the data into little bits and processes it.

Reduce Stage: This level is the result of combining the Shuffle and Reduce stages. The Reducer's job is to take the data from the mapper and process it. It generates a new set of outputs after processing, which will be stored in the HDFS.

Data Node: Data is supplied in advance before any processing takes occur at this node.

Directed Cyclic Graph:A directed cycle graph is a cycle graph with all edges pointing in the same direction.

Message Passing: Message passing is a way for invoking activity (i.e., running a programme) on a computer in computer science. The calling programme delivers a message to a process (which could be an actor or an object), and that process and its supporting infrastructure choose and run relevant code.

Bulk Synchronous Parallel: Bulk Synchronous Parallel (BSP) is a parallel computing programming model and processing framework. The computation is broken down into a series of supersteps. A group of processes running the same code executes concurrently in each superstep and generates messages that are delivered to other processes.

Replication: The process of replicating a double-stranded DNA molecule into two identical DNA molecules is known as DNA replication. Because every time a cell splits, the two new daughter cells must have the same genetic information, or DNA, as the parent cell, replication is required.

Review Questions

Q1: The EXE extension stands for _____

A. executable files
B. extension files
C. extended files
D. None of above

Q2: Select components of distributed file system

A. Term transparency and redundancy
B. Location transparency and redundancy
C. Location transparency and term transparency
D. None of above

Q3: Data replication is a good way to achieve _____ and high concurrency; but it's very hard to maintain frequent changes.

A. fault tolerance
B. detection tolerance
C. both
D. none of above

Q4: Select new computing paradigms

A. RFID
B. Sensor technologies
C. GPS
D. All of the above

Q5: The file type is denoted by _____

A. Filename

B. File identifier
C. File extension
D. None of the mentioned.

Q6: What computer technology is used to describe services and applications that run on a dispersed network using virtualized resources?

A. Distributed Computing
B. Cloud Computing
C. Soft Computing
D. Parallel Computing

Q7: Which one of the following options can be considered as the Cloud?

A. Hadoop
B. Intranet
C. Web Applications
D. All of the mentioned

Q8: The popular models for big data are

A. Directed Acyclic Graph
B. Message Passing
C. Workflow
D. All of above

Q9: The name of three stages in which MapReduce program executes are:

A. Map stage, shuffle stage, and reduce stage
B. shuffle stage, Map stage and reduce stage
C. reduce stage, shuffle stage, and Map stage
D. None of above.

Q10: A directed acyclic graph (DAG) refers to a directed graph which has no _____ cycles.

A. Infinite
B. Directed
C. Direction
D. Noe of above

Q11: Select two important tasks of mapreduce algorithms

A. Map
B. Reduce
C. Both
D. None of above

Q12: The main application areas of Directed Acyclic Graph are

A. Genealogy

B. Citation graphs
C. Job Scheduling
D. All of above

Q13: Full form of BSP is

A. Bulk Synchronous Packets
B. Bald Synchronous Parallel
C. Bulk Switch Parallel
D. Bulk Synchronous Parallel

Q14: Map takes a set of data and converts it into another set of data, where individual elements are broken down into _____

A. Tables.
B. Tuples (key/value pairs).
C. Reduce stage.
D. None of above.

Q15: HDFS stands for

A. Hadoop file system
B. Hadoop flat system
C. Hadoop file switch
D. None of above

Answers for Self Assessment

1.	A	2.	B	3.	A	4.	D	5.	C
6.	B	7.	A	8.	D	9.	A	10.	B
11.	C	12.	D	13.	D	14.	B	15.	A

Review Questions

1. Differentiate between file system and distributed file system.
2. Write down features of Distributed file system.
3. Write down popular models of BIG DATA.
4. Write down challenges of BIG DATA.
5. Write note on the following.
   a. The Age of Internet Computing
   b. High Throughput Computing
6. What are the advantages and disadvantages of distributed file system?

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think. Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier for Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practicesof Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 03: Data Models

CONTENTS

Objectives

- Understand what is data mart
- Understand data format
- Understand data model
- Differentiate between data warehouse and data mart
- Understand what is data stream
- understand streaming sensor data

Introduction

A data mart is a smaller version of a data warehouse that caters to specialised data analysis requirements. It is often derived as a subset of a larger data warehouse. The primary goal of data marts is to do analysis that is difficult to perform in a traditional warehouse due to the varying levels of granularity of data or the need to perform sophisticated computations.

3.1 What is data format?

Applications need to be able to agree some kind of syntax in order to exchange data between them.

Example: If we deliver lecture in French for those students who understand only English. It means protocol has not been followed. Similarly, applications also need to follow some syntax and protocols in order to exchange between them. For this, applications can use standard data format like JSON and XML. Applications need not to agree how data is format and how data is structured. In the same way that routers and switches require standardized protocols in order to communicate, applications need to be able to agree on some kind of syntax in order to exchange data between them.

For this, applications can use standard *data formats* like JSON and XML (among others). Not only do applications need to agree on how the data is formatted, but also on how the data is structured. *Data models* define how the data stored in a particular data format is structured.

Data Format

- A computer programmer typically uses a wide variety of tools to store and work with data in the programs they build.They may use simple variables (single value), arrays (multiple values), hashes (key-value pairs), or even custom objects built in the syntax of the language they're using.Portable format is required.
- Another program may have to communicate with this program in a similar way, and the programs may not even be written in the same language, as is often the case with something like traditional client-server communications as shown in Figure 1.

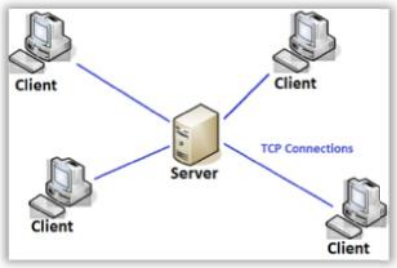


Figure 1 Traditional Client/Server Applications

- This is all perfectly standard within the confines of the software being written. However, sometimes a more abstract, portable format is required. For instance, a non-programmer may need to move data in and out of these programs.
- For example, many third-party user interfaces (UIs) are used to interface with public cloud providers. This is made possible (in a simplified fashion) thanks to standard data formats. The moral of the story is that we need a standard format to allow a diverse set of software to communicate with each other, and for humans to interface with it.

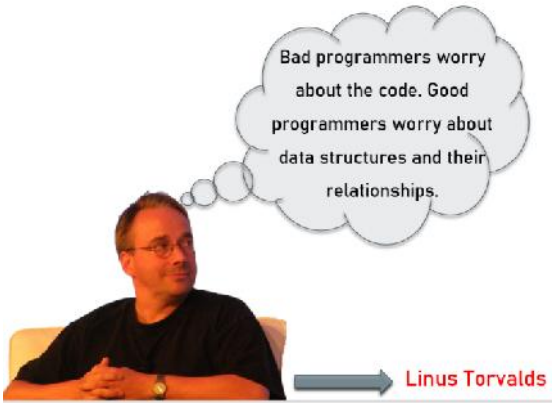
3.2 What is Data Model

In a library, we need to classify all books and arrange them on shelves to make sure we can easily access every book. Similarly, if we have massive amounts of data, we need a system or a method to keep everything in order. The process of sorting and storing data is called "data modeling."A data model is a method by which we can organize and store data. Just as the Dewey Decimal System organizes the books in a library, a data model helps us arrange data according to service, access, and use. Torvalds, the founder of Linux, alluded to the importance of data modeling when he wrote an article on "what makes an excellent programmer":"Poor programmers care about code, and good programmers care about the data structure and the relationships between data." Appropriate models and storage environments offer the following benefits to big data:

Organize and Store Data

Dewey Decimal System organizes the books

- A data model explicitly determines the structure of data.
- Data models are specified in a data modeling (link is external) notation, which is often graphical in form.
- A data model can be sometimes referred to as a data structure (link is external), especially in the context of programming languages (link is external).



How is Data-Model Built?

A data model is built using components that act as abstractions of real-world things.

The simplest data model consists of entities and relationships.As work on the data model progresses, additional detail and complexity are added, including attributes, domains, constraints, keys, cardinality, requirements, relationships—and importantly, definitions of everything in the data model. If we want to understand the data we have—and how to use it—a foundational model is required.

3.3 Benefits of Appropriate Models and Storage Environments to Big Data

- P**erformance**: Good data models can help us quickly query the required data and reduce I/O throughput.
- **Cost:** Good data models can significantly reduce unnecessary data redundancy, reuse computing results, and reduce the storage and computing costs for the big data system.
- **Efficiency:** Good data models can greatly improve user experience and increase the efficiency of data utilization.
- **Quality:** Good data models make data statistics more consistent and reduce the possibility of computing errors.

Therefore, it is without question that a big data system requires high-quality data modeling methods for organizing and storing data, allowing us to reach the optimal balance of performance, cost, efficiency, and quality.

Tips for creating effective big data models

Data modeling is a complex science that involves organizing corporate data so it fits the needs of business processes. It requires the design of logical relationships so the data can interrelate with each other and support the business. The logical designs are then translated into physical models that consist of storage devices, databases and files that house the data. Historically, businesses have used relational database technology like SQL to develop data models because it is uniquely suited for flexibly linking dataset keys and data types together in order to support the informational needs of business processes. Unfortunately, big data, which now comprises a large percentage of data under management, does not run on relational databases. It runs on non-relational databases like NoSQL. This leads to the belief that you don't need a model for big data.The problem is, you do need data modeling for big data.

Don't try to impose traditional modeling techniques on big data

Traditional, fixed record data is stable and predictable in its growth. This makes it relatively easy to model. In contrast, big data's exponential growth is unpredictable, as are its myriad forms and sources. When sites contemplate modeling big data, the modeling effort should center on constructing open and elastic data interfaces, because you never know when a new data source or form of data could emerge. This is not a priority in the traditional fixed record data world.

Design a system, not a schema

- In the traditional data realm, a relational database schema can cover most of the relationships and links between data that the business requires for its information support. This is not the case with big data, which might not have a database, or which might use a database like NoSQL, which requires no database schema.Because of this, big data models should be built on systems, not databases. The system components that big data models should contain are business information requirements, corporate governance and security, the physical storage used for the data, integration and open interfaces for all types of data, and the ability to handle a variety of different data types.

Look for big data modeling tools

There are commercial data modeling tools that support Hadoop, as well as big data reporting software like Tableau . When considering big data tools and methodologies, IT decision makers should include the ability to build data models for big data as one of their requirements.

Focus on data that is core to your business

Mountains of big data pour into enterprises every day, and much of this data is extraneous. It makes no sense to create models that include all data. The better approach is to identify the big data that is essential to your enterprise, and to model that data.

Deliver quality data

Superior data models and relationships can be effected for big data if organizations concentrate on developing sound definitions for the data and thorough metadata that describes where the data came from, what its purpose is, etc. The more you know about each piece of data, the more you can place it properly into the data models that support your business.

Look for key inroads into the data

One of the most commonly used vectors into big data today is geographical location. Depending on your business and your industry, there are also other common keys into big data that users want. The more you can identify these common entry points into your data, the better you will be able to design data models that support key information access paths for your company.

3.4 What is data mart?

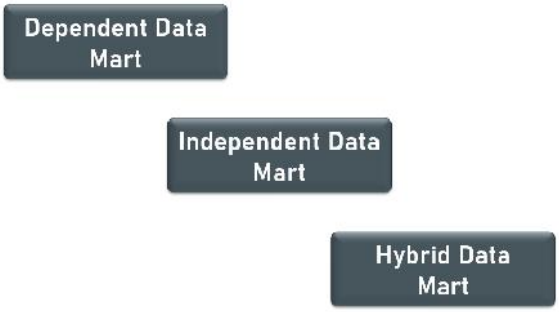
It is a data store which is designed for a particular department of an organization, or data mart is a subset of Datawarehouse that is usually oriented to a specific purpose.A data mart is a subset of a data warehouse oriented to a specific business line. Data marts contain repositories of summarized data collected for analysis on a specific section or unit within an organization, for example, the sales department. A data warehouse is a large centralized repository of data that contains information

from many sources within an organization. The collated data is used to guide business decisions through analysis, reporting, and data mining tools.

Reasons for using data mart

- Easy access of frequent data.
- Improved end user response time.
- Easy creation of data marts.
- Less cost in building the data mart.

3.5 Different types of data mart



Dependent data mart

- In this the data mart is built by drawing data from central data warehouse that already exists.A dependent data mart allows sourcing organization's data from a single Data Warehouse. It is one of the data martsexamples which offers the benefit of centralization. If you need to develop one or more physical data marts, then you need to configure them as dependent data marts.
- Dependent data mart can be built in two different ways
    - Either where a user can access both the data mart and data warehouse, depending on need, or where access is limited only to the data mart.
    - The second approach is not optimal as it produces sometimes referred to as a data junkyard. In the data junkyard, all data begins with a common source, but they are scrapped, and mostly junked.

Independent data mart

In this, the data mart is built by drawing from operational or external sources of data or both.

Hybrid Data Mart

A hybrid data mart combines input from sources apart from Data warehouse. It is the best data mart example suited for multiple database environments and fast implementation turnaround for any organization. It also requires least data cleansing effort. Hybrid Data mart also supports large storage structures, and it is best suited for flexible for smaller data-centric applications.

Three types of datamart are shown in Figure 2.

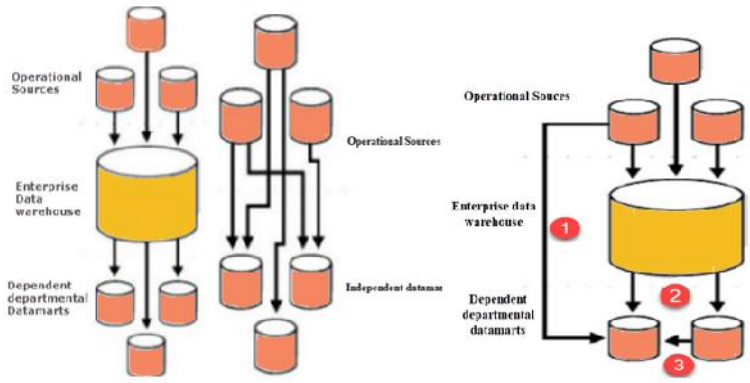


Figure 2 Independent, dependent and hybrid data mart

Advantages of Data mart

Simpler, more focused & flexible.

Low cost for both h/w and s/w.

Faster and cheaper to build.

Stores data closer that enhances performance.

Disadvantages of data mart

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a big hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

3.6 Difference between Data warehouse and Data mart

Data Warehouse	Data Mart
Data warehouse is a Centralised system.	While it is a Decentralised system
In data warehouse, lightly denormalization takes place.	While in Data mart, highly denormalization takes place
Data warehouse is top-down model.	While it is a bottom-up model.
To built a warehouse is difficult.	While to build a mart is easy
In data warehouse, Fact constellation schema is used.	While in this, Star schema and snowflake schema are used.

Data Warehouse	Data Mart
Data warehouse is a Centralised system.	While it is a Decentralised system
In data warehouse, lightly denormalization takes place.	While in Data mart, highly denormalization takes place
Data warehouse is top-down model.	While it is a bottom-up model.
To built a warehouse is difficult.	While to build a mart is easy
In data warehouse, Fact constellation schema is used.	While in this, Star schema and snowflake schema are used.

3.7 What Does Big Data Streaming Mean

- Big data streaming is a process in which big data is quickly processed in order to extract real-time insights from it. The data on which processing is done is the data in motion. Big data streaming is ideally a speed-focused approach wherein a continuous stream of data is processed as shown in Figure 3

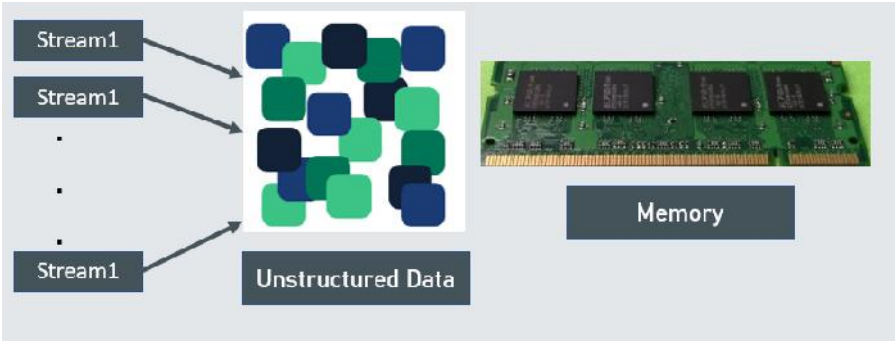


Figure 3 Data streaming

3.8 Data Streaming

- Big data streaming is a process in which large streams of real-time data are processed with the sole aim of extracting insights and useful trends out of it. A continuous stream of unstructured data is sent for analysis into memory before storing it onto disk. This happens across a cluster of servers. Speed matters the most in big data streaming. The value of data, if not processed quickly, decreases with time.With the explosion of data in recent years, there's more emphasis on data-based decision-making for all companies. But what if we could process data and act on it in real-time? What if we could be proactive instead of reactive to improve performance? Streaming data and streaming analytics now make that possible for almost every company in every industry to drive intelligence and action in real-time. And with an accelerated push for automation and digitization in a post-coronavirus world, it is clear that those companies that leverage real-time data are much more likely to gain a competitive edge on their competitors.
- Speed matters the most in big data streaming.
- The value of data, if not processed quickly, decreases with time.
- Real-time streaming data analysis is a single-pass analysis.
- Analysts cannot choose to reanalyze the data once it is streamed.

- Dynamic data that is generated continuously from a variety of sources is considered streaming data. Whether that data comes from social media feeds or sensorsFigure

4 and cameras, each record needs to be processed in a way that preserves its relation to other data and sequence in time.Log files, e-commerce purchases, weather events, utility service usage, geo-location of people and things, server activity, and more are all examples where real-time streaming data is created.When companies are able to analyze streaming data they receive, they can get real-time insights to understand exactly what is happening at any given point in time. This enables better decision-making as well as provide customers with better and more personalized services. Nearly every company is or can use streaming data.

Figure 4 Social media and sensors

3.9 Use Cases for Real-Time and Streaming Data

Any data-driven organization, which today is nearly all of them, can use real-time and streaming data to improve results. Here are just a few use cases across multiple industries:

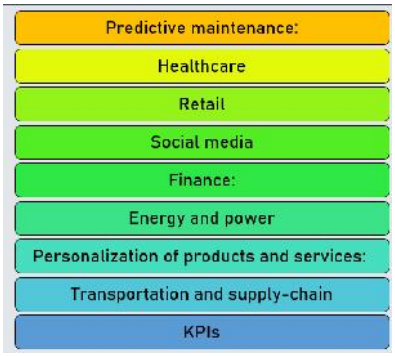


Figure 5 Use cases for real time streaming data

Predictive Maintenance

When companies can identify maintenance issues prior to breakdowns or system failure, they will save time, money, and other potentially catastrophic effects on the business. Any **company that has equipment of any kind that has sensors** or cameras—again, that's most equipment these days—will create streaming data. From monitoring the performance of **trucks**, and **airplanes**, to predicting issues with **complex manufacturing equipment**, real-time data and analytics is becoming critical to modern enterprises today.

Figure 6 Predictive maintenance

Healthcare

Just like in a manufacturing environment, wearables, and healthcare equipment such as glucometers, connected scales, heart rate and **blood pressure monitors** have sensors that monitor a patient's vitals and essential body functions. This equipment is also crucial for effective remote

patient monitoring that supports clinicians who don't have the bandwidth to be everywhere all the time. It's literally a matter of life or death. Immediate insights can improve patient outcomes and experiences.

Retail

Real-time data streaming from IoT sensors and video are driving a modern **retail renaissance**. Brick-and-mortar retail stores can engage customers in the moment thanks to streaming data. Location-based marketing, trend insights, and improvements to operational efficiency, such as product movement or product freshness, are all possible with real-time insights. Understanding what a consumer wants when they want it "in the moment" is not only valuable in retail. Any company that is able to understand and respond immediately to what its customer wants in **micro-moments** will have a better chance of being successful, whether it's to deliver something a consumer wants to learn, discover, watch or buy.

Social media

With cries of "fake news" and instances of social media bullying continuing to rise, the need for real-time monitoring of posts to quickly take action on offensive and "**fake news**" is more important than ever. Under mounting pressure, **social media platforms** are creating tools to be able to process the huge volume of data created quickly and efficiently to be able to take action as immediately as possible, especially to prevent bullying.

Finance

On the trading floor, it's easy to see how understanding and acting on information in real-time is vital, but streaming data also helps the financial functions of any company by processing transactional information, identify fraudulent actions, and more. For example, **MasterCard** is using data and analytics to helping financial organizations quickly and easily identify fraudulent merchants to reduce risk. Similarly, by gaining the ability to process real-time data, **Rabobank** is able to detect warning signals in extremely early stages of where clients may go into default.

Energy and Power

In the **energy sector**, companies are working to optimize fossil fuels and adopt more sustainable power systems. A continuous flow of data helps with predictive maintenance on equipment as well as to better understand consumer demand and improve business and operations.

Personalization of products and services

Companies can better respond to consumers' demands to have what they want (and even what they don't know they want yet) thanks to streaming data. From online news publications that serve up content a particular reader is most interested in to streaming services that recommend the next things to watch, personalization adds value to the customer experience but is only possible in real-time because of streaming data.

Transportation and supply-chain

Streaming data powers the **internet of trains**, makes connected and autonomous vehicles possible and safer, and is crucial in making fleet operations more efficient.

KPIs

Leaders can make decisions based on real-time KPIs such as financial, customer, or operational performance data. Previously, this analysis was reactive and looked back at past performance. Today, real-time data can be compared with historical information to give leaders a perspective on business that informs real-time decisions.As you can see, streaming data is increasingly important to most companies in most industries. Successful companies are integrating streaming analytics to move their data analytics from a reactive to a more proactive real-time approach. The best ones will be thinking about integrating their real-time data with predictive models and scenario analysis to gain strategic foresight.However, in order to harness fast and streaming data, organizations today need an end-to-end data management and analytics platform that can collect, process, manage, and analyze data in real-time to drive insights and enable machine learning to implement some of the most compelling use cases. Most importantly, they need to be able to do these with the robust security, governance, data protection, and management capabilities that enterprises require.

3.10 Data Lake

A data lake is a centralized repository that allows you to store all your structured and unstructured data at any scale. You can store your data as-is, without having to first structure the data, and run different types of analytics—from dashboards and visualizations to big data processing, real-time analytics, and machine learning to guide better decisions as shown in Figure 7.

Figure 7 Data Lake

Why do you need a data lake?

Organizations that successfully generate business value from their data, will outperform their peers. An Aberdeen survey saw organizations who implemented a Data Lake outperforming similar companies by 9% in organic revenue growth. These leaders were able to do new types of analytics like machine learning over new sources like log files, data from click-streams, social media, and internet connected devices stored in the data lake.This helped them to identify, and act upon opportunities for business growth faster by attracting and retaining customers, boosting productivity, proactively maintaining devices, and making informed decisions.

3.11 DataLake vs.Data warehouse

Depending on the requirements, a typical organization will require both a data warehouse and a data lake as they serve different needs, and use cases.A data warehouse is a database optimized to analyze relational data coming from transactional systems and line of business applications. The data structure, and schema are defined in advance to optimize for fast SQL queries, where the results are typically used for operational reporting and analysis. Data is cleaned, enriched, and transformed so it can act as the "single source of truth" that users can trust.

Characteristics	Data Lakes	Datawarehouse
Data	Relational from transactional systems, operational databases, and line of business applications	Non-relational and relational from IoT devices, web sites, mobile apps, social media, and corporate applications
Schema	Designed prior to the DW implementation (schema-on-write)	Written at the tme of analysis (schema-on-read)
Price/Performance	Fastest query results using higher cost storage	Query results getting faster using low-cost storage
Data Quality	Highly curated data that serves as the central version of the truth	Any data that nay or may not be curated (ie raw data)
Users	Business analysts	Data scientists Data developers, and Business analysts (usingcurated data)
Analytics	Batch reporting, BI and visualizations	Machine Learning, Predictive analytics, datadiscovery and profiling

3.12 The essential elements of a Data Lake and Analytics solution

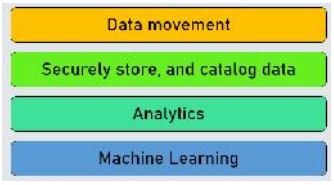


Figure 8 Essential elements of a data lake

Data movement

Data Lakes allow you to import any amount of data that can come in real-time. Data is collected from multiple sources, and moved into the data lake in its original format. This process allows you to scale to data of any size, while saving time of defining data structures, schema, and transformations.

Securely store, and catalog data

Data Lakes allow you to store relational data like operational databases and data from line of business applications, and non-relational data like mobile apps, IoT devices, and social media. They also give you the ability to understand what data is in the lake through crawling, cataloging, and indexing of data. Finally, data must be secured to ensure your data assets are protected.

Analytics

Data Lakes allow various roles in your organization like data scientists, data developers, and business analysts to access data with their choice of analytic tools and frameworks. This includes open source frameworks such as Apache Hadoop, Presto, and Apache Spark, and commercial offerings from data warehouse and business intelligence vendors. Data Lakes allow you to run analytics without the need to move your data to a separate analytics system.

Machine Learning

Data Lakes will allow organizations to generate different types of insights including reporting on historical data, and doing machine learning where models are built to forecast likely outcomes, and suggest a range of prescribed actions to achieve the optimal result.

3.13 The value of a Data Lake

The ability to harness more data, from more sources, in less time, and empowering users to collaborate and analyze data in different ways leads to better, faster decision making. Examples where Data Lakes have added value include as show in Figure 9:

Figure 9 The value of a data lake

Improved customer interactions

A Data Lake can combine customer data from a CRM platform with social media analytics, a marketing platform that includes buying history, and incident tickets to empower the business to understand the most profitable customer cohort, the cause of customer churn, and the promotions or rewards that will increase loyalty.

Improve R&D innovation choices

A data lake can help your R&D teams test their hypothesis, refine assumptions, and assess results — such as choosing the right materials in your product design resulting in faster performance, doing genomic research leading to more effective medication, or understanding the willingness of customers to pay for different attributes.

Increase operational efficiencies

The Internet of Things (IoT) introduces more ways to collect data on processes like manufacturing, with real-time data coming from internet connected devices. A data lake makes it easy to store, and run analytics on machine-generated IoT data to discover ways to reduce operational costs, and increase quality.

3.14 The challenges of Data Lakes

The main challenge with a data lake architecture is that raw data is stored with no oversight of the contents. For a data lake to make data usable, it needs to have defined mechanisms to catalog, and secure data. Without these elements, data cannot be found, or trusted resulting in a "data swamp." Meeting the needs of wider audiences require data lakes to have governance, semantic consistency, and access controls.

Figure 10 Challenges of a data lake

3.15 Streaming sensor data

The increasing availability of cheap, small, low-power sensor hardware and the ubiquity(omnipresence) of wired and wireless networks has led to the prediction that 'smart environments' will emerge(arise) in the near future. Often these systems will be monitoring conditions in the real world: weather, temperature, road traffic, location of objects, prices on the stock market. In some cases, a regular update (where 'regular' could range from milliseconds to hours) about these conditions is produced; in others, the system gives notifications when special changes occur. The sensors in these environments collect detailed information about the situation people are in, which is used to enhance information-processing applications that are present on their mobile and 'ambient(existing or present on all sides)' devices. Bridging the gap between sensor data and application information poses new requirements to data management.

Supply side: sensors

The supply side of a smart environment consists of a myriad of sensors that produce data at possibly very high rates. The PocketLab One is designed to be a basic starter solution (and would fit the needs of most science classes). It can measure motion, acceleration, angular velocity, magnetic field, pressure, altitude, and temperature. The use of sensors for inserting data into the system has several consequences. The values will not exactly correspond to the real world due to measurement system errors, noise and uncontrolled conditions as shown in

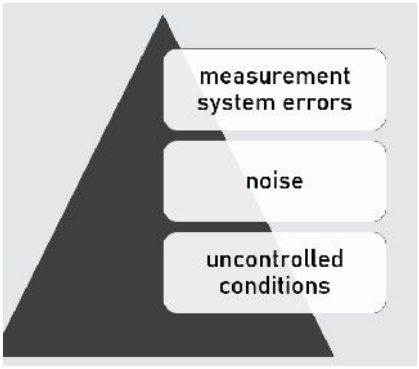


Figure 11 Supply side sensors

The distributed, wireless and battery-powered nature of sensor networks will force data management to take sensor failure, network latency and loss into account. At the other hand, there will be a lot of redundant (or, in statistical terms, highly correlated) data to counter these negative features. A couple of remarks to sketch the situation. At the other hand, there will be a lot of redundant (or, in statistical terms, highly correlated) data to counter these negative features. A couple of remarks to sketch the situation.

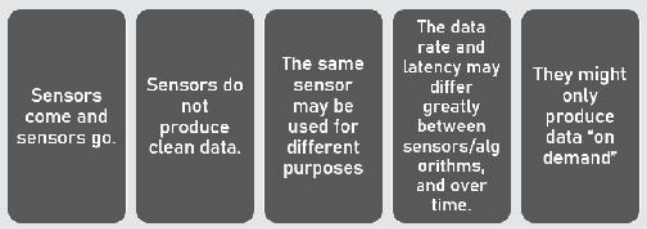


Figure 12 A couple of remarks to sketch the situation

- Sensors come and sensors go. They can fail because their battery runs out, and start up again when it is replaced. They can be disconnected, moved and connected at a different place. They can be replaced altogether by a newer model. They can have wireless connections which do not work all the time.
- Sensors do not produce clean data. Averages have to be taken, noise filters have to be applied, environmental influences (e.g.echos) have to be accounted for.
- The same sensor may be used for different purposes. Different algorithms are applied on the raw data depending on what you want to know, e.g. using a microphone for speaker identification, speaker positioning or estimation of the environmental noise level.
- The data rate and latency may differ greatly between sensors/algorithms, and over time: In some cases, it may be parameterizable (i.e. a sensor or algorithm can be configured to produce output at several rates). In some cases, the term "data rate" might not even apply at all (e.g. RFID readers which produce a reading (or a burst of readings) whenever a tag is detected).
- They might only produce data "on demand" because of the cost associated with it. This cost may be power, but it may also be money if the sensor belongs to another party (think of weather or traffic sensors).

Demand side: applications

Applications are typically uninterested in details about the sensing architecture and will need a sufficiently high-level 'world model' to base their behavior on. When applications are connected, they will probably be interested in high-resolution, live data; when they have been disconnected for a while they might rather request a summary (e.g. properties like average, variation, frequencies). Some other characteristics:

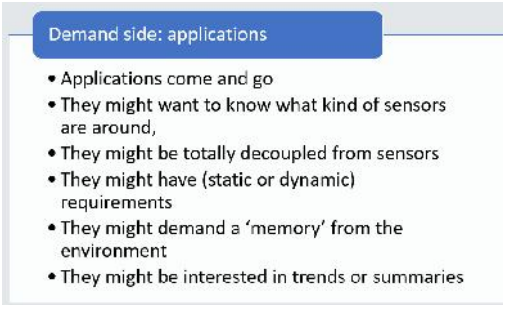


Figure 13 Demand side: applications

- Applications come and go. They can be turned on and off at will; they are duplicated for each new user; they are upgraded. They are disconnected at one place and connected at another, and might be interested in what happened in the meantime.
- They might want to know what kind of sensors are around, and adapt their information demands to this.

- They might be totally decoupled from sensors, and just want to know e.g. which person is at a certain desk.
- They might have (static or dynamic) requirements about the rate at which data is delivered to them. This rate may vary greatly from application to application.
- They might demand a 'memory' from the environment to discover details of specific events in the past.
- They might be interested in trends or summaries rather than in specifics.

3.16 Uses of sensor data

Sensor data is ubiquitous and may be applied to a variety of sectors, including health care, weather forecasting, sound analysis, and video streaming. Let's look at some of the applications of sensor data.

Health sensor data:

Many medical facilities keep track of patients' vital signs in real time. Heart rate, electrodermal activity, brain waves, temperature, and other vital indicators are among them. When gathered, real-time sensor data is vast and may be classified as big data. ICT technology can utilise this information to diagnose patients.

Weather data

Many satellites provide real-time weather data streaming in order to capture critical signals for the weather. This information is used to forecast the weather.

Sensor data from IOT

Custom internet of things devices, such as Raspberry Pi, Apple watches, smartphones, health watches, and so on, may collect a lot of sensor data. This data can be transmitted in real time, allowing certain servers to access it.

Summary

- A data mart is a structure / access pattern used to get client-facing data in data warehouse setups. A data mart is a subset of a data warehouse that is often focused on a single business line or team.
- A data mart is a subset of a data warehouse that is focused on a certain business line, department, or topic area. Data marts make specialised data available to a designated group of users, allowing them to rapidly obtain key insights without having to sift through a whole data warehouse.
- A dependent data mart enables data from several organisations to be sourced from a single Data Warehouse. It is an example of a data mart that provides the benefit of centralization. You must setup one or more physical data marts as dependent data marts if you need to create them.
- Without the usage of a central data warehouse, an independent data mart is formed. This type of Data Mart is best suited for smaller groups inside a company.
- A data lake is a system or repository that stores data in its original/raw form, which is often object blobs or files.
- Data that is continually created by several sources is referred to as streaming data. Without having access to all of the data, such data should be handled sequentially utilising stream processing techniques.

Keywords

Predictive maintenance: Predictive maintenance is the application of data-driven, proactive maintenance approaches to examine equipment status and anticipate when repair should be conducted.

Dependent data marts: An enterprise data warehouse is used to establish a reliant data mart. It's a top-down technique that starts with keeping all company data in one single area and then extracting a clearly defined piece of the data for analysis as needed.

Independent data marts: An independent data mart is a stand-alone system that concentrates on a single topic area or business activity without the usage of a data warehouse. Data is retrieved from internal or external data sources (or both), processed, and then deposited into a data mart repository, where it is kept until it is required for business analytics.

Hybrid data marts: Data from an existing data warehouse and other operational source systems is combined in a hybrid data mart. It combines the speed and end-user emphasis of a top-down strategy with the benefits of the bottom-up method's enterprise-level integration.

Maintenance: In industrial, commercial, and residential settings, maintenance include functioning inspections, servicing, repairing, or replacing essential devices, equipment, machinery, building structures, and supporting utilities.

Data Lake:A data lake is a storage repository that stores a large amount of raw data in its original format until it is required for analytics applications. A data lake differs from a standard data warehouse in that it stores data in flat architecture, mostly in files or object storage, rather than hierarchical dimensions and tables.

Data ware house: A data warehouse is a huge collection of corporate data used to aid decision-making inside a company. The data warehouse idea has been around since the 1980s, when it was created to aid in the transfer of data from being used to power operations to being used to feed decision support systems that disclose business insight.

Self Assessment

Q1: Which of the following is an aim of data mining?

A. To explain some observed event or condition
B. To analyze data for expected relationships
C. To confirm that data exists
D. To create a new data warehouse
E.

Q2: Which of the following is a data warehouse?

A. focused on a single subject
B. Organized around important subject areas
C. Collection of computers
D. None of above

Q3: Select types of data mart.

A. Dependent datamart
B. Independent datamart
C. Hybrid datamart
D. All of the above

Q4: Select statement which is not true about data warehouse

A. Data warehouse is flexible
B. Data warehouse has long life
C. Data warehouse is top-down model.
D. Data warehouse is a de-centralized system

Q5: _____ is built by drawing data from central data warehouse that already exists.

A. Dependent datamart
B. Independent datamart
C. Hybrid datamart
D. All of the above

Q6: _____ is built by drawing from operational or external sources of data or both.

A. Dependent datamart
B. Independent datamart
C. Hybrid datamart
D. All of the above

Q7: A _____ data mart combines input from sources apart from Data warehouse

A. Dependent datamart
B. Independent datamart
C. Hybrid datamart
D. All of the above

Q8: Big data streaming is a process in which big data is quickly processed in order to extract _____ insights from it.

A. real-time
B. streaming data
C. both a and b
D. None of above

Q9: Dynamic data that is generated continuously from a variety of sources is considered _____

A. real-time
B. steaming data
C. both a and b
D. None of above

Q10: _____ is using data and analytics to helping financial organizations quickly and easily identify fraudulent merchants to reduce risk.

A. Debit Card
B. Credit Card
C. MasterCard
D. None of the above

Q11: Which of the following data streaming assertions is correct?

A. Stream data is inherently unstructured.
B. Stream data has a fast rate of change.
C. Stream components are not able to be saved to disc.
D. None of above

Q12: The increasing availability of cheap, small, low-power sensor hardware and has led to the prediction that _____ will arise in the near future.

A. Small-environment
B. Supply side
C. both a and b
D. None of above

Q13: The _____ of a smart environment consists of a myriad of sensors that produce data at possibly very high rates real-time

A. streaming data
B. supply side
C. both a and b
D. None of above

Q14: The _____ is designed to be a basic starter solution

A. streaming data
B. supply side
C. PocketLab
D. None of the above

Q15: _____ can fail because their battery runs out, and start up again when it is replaced

A. Sensors come and sensors go
B. Sensors do not produce clean data
C. Both
D. None of above

Answers for Self Assessment

1.	A	2.	B	3.	D	4.	D	5.	A
6.	B	7.	C	8	A	9	B	10	C
11	B	12	A	13	B	14	C	15	A

Review Questions

1. Difference between data mart and data ware house.
2. Write down the Tips for Creating Effective Big Data Models.
3. Explain different types of data mart.
4. Write down advantages and disadvantages of data mart.
5. What do you understand by data streaming? Explain Use Cases for Real-Time and Streaming Data.

Further Readings

Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.

Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.

McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com

Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practice of Scalable Realtime Data Systems. Manning Publications.

Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 04: NOSQL Management

CONTENTS

Objectives

- identify key differences between NOSQL and relational databases
- appreciate the architecture and types of NOSQL databases
- describe the major types of NOSQL databases and their features
- learn distribute data models
- learn hadoop partitioner

Introduction

- A NOSQL database is a clever way of cost-effectively organizing large amounts of heterogeneous data for efficient access and updates. An ideal NOSQL database is completely aligned with the nature of the problems being solved, and is superfast in accomplishing that task.This is achieved by relaxing many of the integrity and redundancy constraints of storing data in relational databases. Data is thus stored in many innovative formats closely aligned with business need. The diverse NOSQL databases will ultimately collectively evolve into a holistic set of efficient and elegant knowledge stored at the heart of a cosmic computer.

4.1 What is Relational Database

Relational data management systems (RDBMs) are a powerful and universally used database technology by almost all enterprises. Relational databases are structured and optimized to ensure accuracy and consistency of data, while also eliminating any redundancy of data.These databases

are stored on the largest and most reliable of computers to ensure that the data is always available at a granular level and at a high speed.

Why NOSQL Database Emerge?

Big data is, however, a much larger and unpredictable stream of data. Relational databases are inadequate for this task, and will also be very expensive for such large data volumes. Managing the costs and speed of managing such large and heterogeneous data streams requires relaxing many of the strict rules and requirements of relational database. Depending upon which constraint(s) are relaxed, a different kind of database structure will emerge. These are called NOSQL databases, to differentiate them from relational databases that use Structured Query Language (SQL) as the primary means to manipulate data.

NOSQL databases are next-generation databases that are non-relational in their design. The name NOSQL is meant to differentiate it from antiquated, 'pre-relational' databases.Today, almost every organization that must gather customer feedback and sentiments to improve their business, uses a NOSQL database. NOSQL is useful when an enterprise needs to access, analyze, and utilize massive amounts of either structured or unstructured data that's stored remotely in virtual servers across the globe.

- NOSQL database is useful when

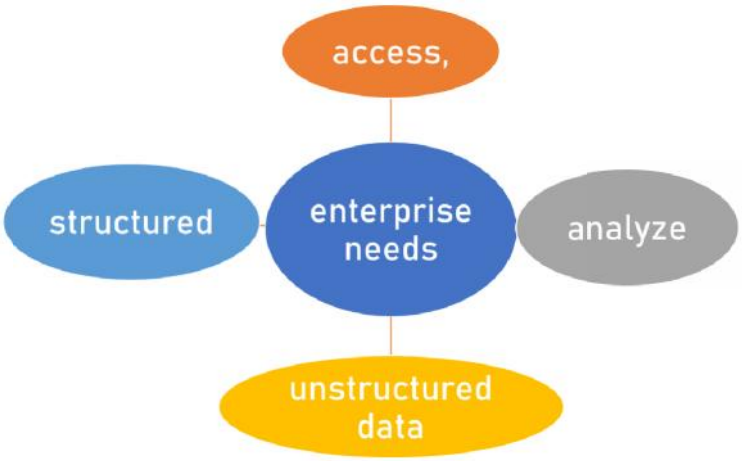


Figure 1: NOSQL database

- The constraints of a relational database are relaxed in many ways. For example, relational databases require that any data element could be randomly accessed and its value could be updated in that same physical location. However, the simple physics of storage says that it is simpler and faster to read or write sequential blocks of data on a disk. Therefore, NOSQL database files are written once and almost never updated in place. If a new version of a part of the data becomes available, it would be appended to the respective files. The system would have the intelligence to link the appended data to the original

file.These differ from each other in many ways. First, NOSQL databases do not support relational schema or SQL language. The term NOSQL stands mostly for "Not only SQL".

- Second, their transaction processing capabilities are fast but weak, and they do not support the ACID (Atomicity, Consistency, Isolation, Durability) properties associated with transaction processing using relational databases. Instead, they support BASE properties (Basically Available, Soft State, and Eventually Consistent). NOSQL databases are thus approximately accurate at any point in time, and will be eventually consistent. Third, these databases are also distributed and horizontally scalable to manage web-scale databases using Hadoop clusters of storage. Thus, they work well with the write-once and read-many storage mechanism of Hadoop clusters. Table 6.1 lists comparative features of RDBMS and NOSQL.

4.2 RDBMS vs. NOSQL

Third, these databases are also distributed and horizontally scalable to manage web-scale databases using Hadoop clusters of storage.

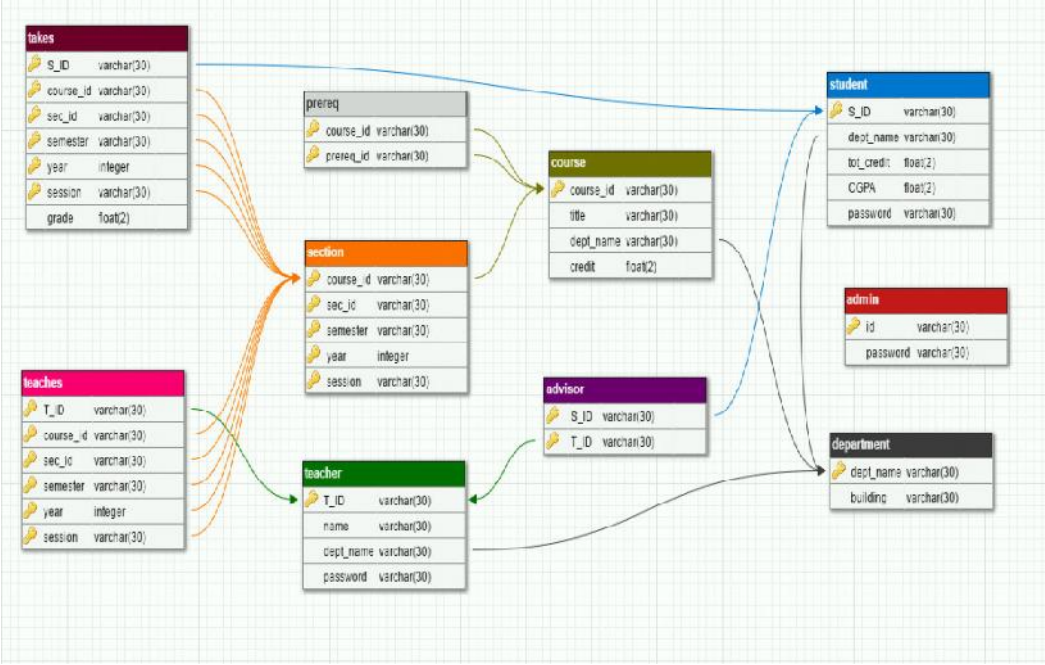


Figure 2: Do not support relational schema

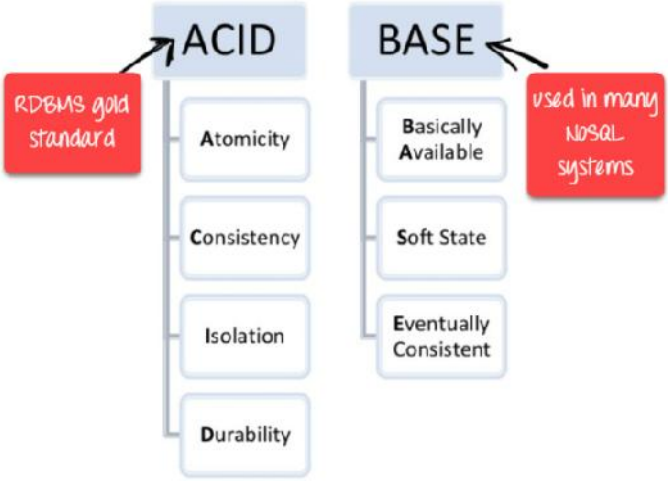


Figure 3: RDBMS vs NOSQL

4.3 Types of NOSQL Databases

The variety of big data means that file size and types will vary enormously. Despite the name, a NOSQL database does not necessarily prohibit structured query language (like SQLWhile some of the NOSQL systems are entirely non-relational, others just avoid some selected functionality of RDMS such as fixed table schemas and join operations. For NOSQL systems, instead of using tables, the data can be organized the data in key/value pair format, and then SQL can be used. There are specialized NOSQL databases to suit different purposes.The choice of NOSQL database depends on the system requirements. There are at least 200 implementations of NOSQL databases of these four types. Visit NOSQL-database. org for more. As you can see in the

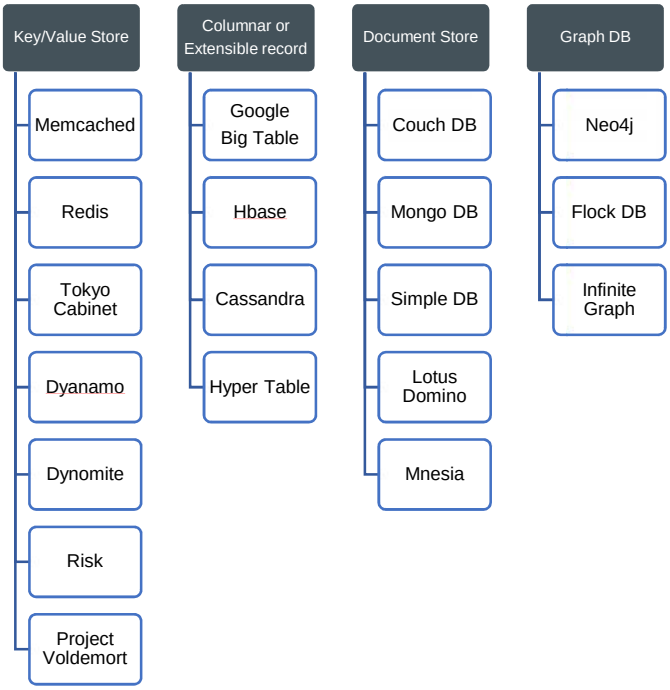


Figure 4. Here are some recent offerings in these categories.

Table 1 Comparative features of RDBMS and NOSQL.

Feature	RDBMS	NOSQL
Applications	Mostly centralized Applications (e.g. ERP)	Mostly designed for the decentralized applications (e.g. Web, mobile, sensors)
Rigor	Support ACID properties for Transaction Processing	Support BASE properties for approximate Reporting
Applications	Mostly centralized Applications (e.g. ERP)	Mostly designed for the decentralized applications (e.g. Web, mobile, sensors)
Rigor	Support ACID properties for Transaction Processing	Support BASE properties for approximate Reporting
Availability	Moderate to high	Continuous availability to receive and serve data
Velocity	Moderate velocity of data	High velocity of data (devices, sensors, social media, etc.). Low latency of access
Data Volume	Moderate size; archived after for a certain period	Huge volume of data, stored mostly for a long time or forever; Linearly scalable DB.
Data Sources	Data arrives from one or few, mostly predictable sources	Data arrives from multiple locations and are of unpredictable nature
Data type	Data are mostly structured	Structured or unstructured data
Data Access	Primary concern is reading the data	Concern is both read and write
Technology	Standardized relational schemas; SQL language	Many designs with many implementations of data structures and access languages
Cost	Expensive; commercial	Low; open-source software

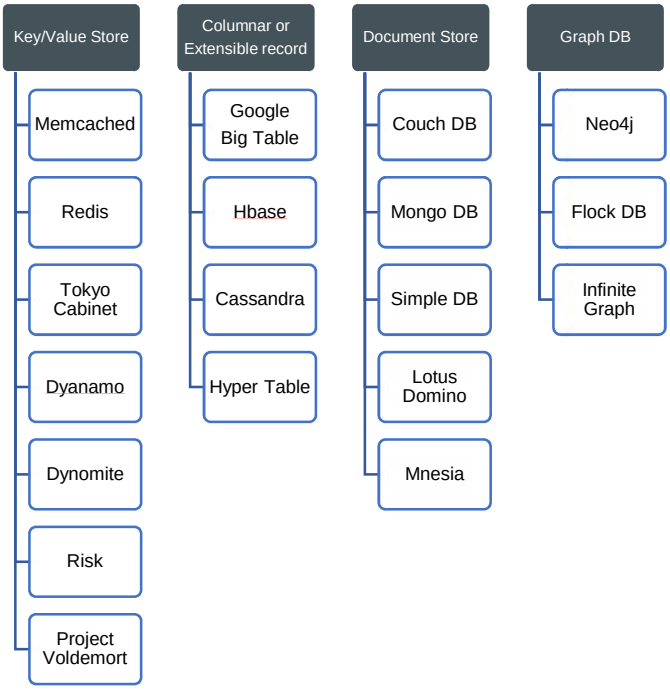


Figure 4: Offerings in NOSQL databases

Columnar Databases:

These are database structures that include only the relevant columns of the dataset, along with the key-identifying information. These are useful in speeding up some oft-sought queries from very large data sets. Suppose there is an extremely large data warehouse of web log access data, which is rolled up by the number of web access by the hour. This needs to be queried, or summarized often, involving only some of the data fields from the database. Thus the query could be speeded up by organizing the database in a columnar format. This is useful for content management systems, blogging platforms, maintaining counters, expiring usage, heavy write volume such as log aggregation. Column family databases for systems well when the query patterns have stabilized.

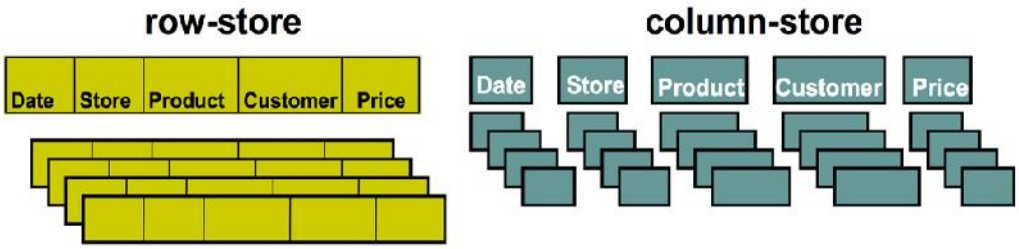


Figure 5: Columnar databases

HBase and Cassandra are the two of the more popular Columnar database offerings. HBase was developed at Yahoo, and comes as part of the Hadoop ecosystem. Cassandra was originally developed at Facebook to serve its exponentially growing user base, which is now close to 2 billion people. It was open sourced in 2008.

Key/Value Pair Databases

There could be a collection of many data elements such as a collection of text messages, which could also fit into a single physical block of storage. Each text message is a unique object. This data would need to be queried often. That collection of messages could also be stored in a key-value pair format, by combining the identifier of the message and the content of the message. Key-value databases are useful for storing session information, user profiles, preferences, and shopping cart data. Key-value databases do not work so well when we need to query by non-key fields or on multiple key fields at the same time.Dynamo is a NOSQL highly available key-value structured

storage system that has properties of both databases and distributed hash tables. Amazon DynamoDB is a fully managed NOSQL database service that provides fast and predictable performance with seamless scalability.

DynamoDB automatically spreads the data and traffic for your tables over enough servers to handle your throughput and storage requirements, while maintaining consistent and fast performance.

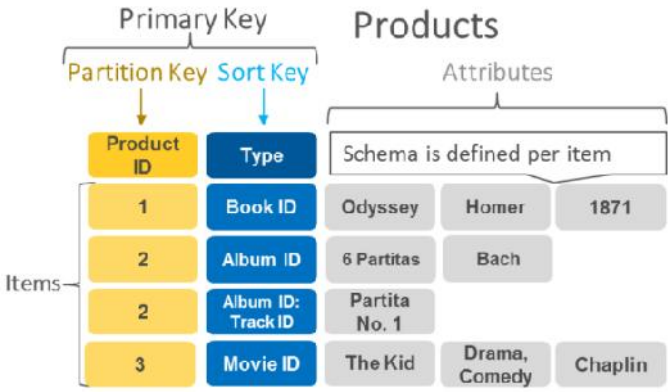


Figure 6:Key-Value Pair databases

Document Database

These databases store an entire document of any size, as a single value for a key element. Suppose one is storing a 10GB video movie file as a single object. An index could store the identifying information about the movie, and the address of the starting block. The system could handle the rest of storage details. This storage format would be a called document store format. Document databases are generally useful for content management systems, blogging platforms, web analytics, real-time analytics, ecommerce-applications. Document databases would not be useful for systems that need complex transactions spanning multiple operations or queries against varying aggregate structures.

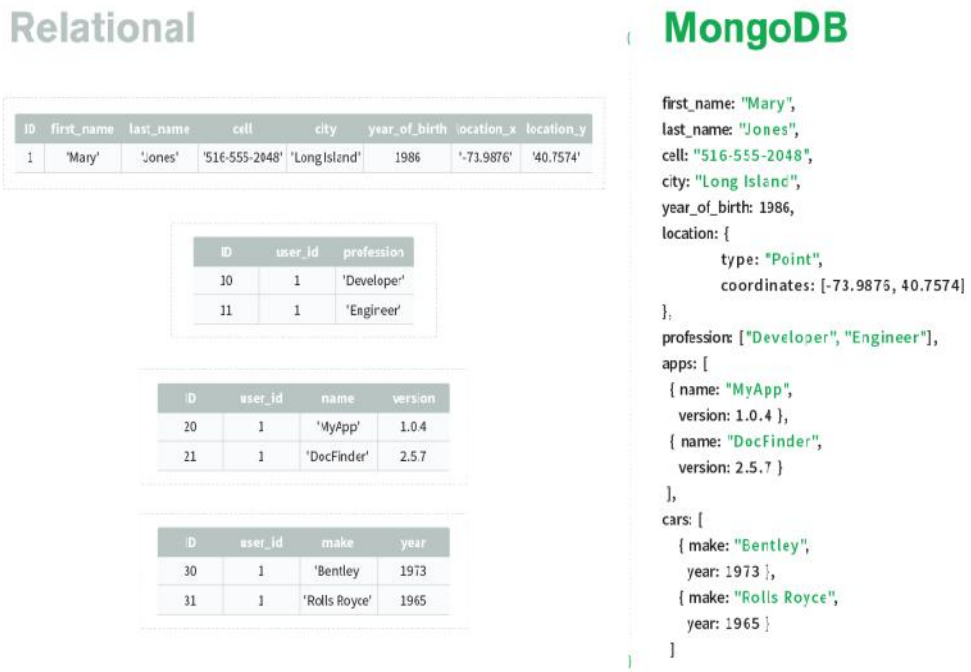


Figure 7: Document Database

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling. A record in MongoDB is a document, which is a data structure composed of

field and value pairs. The values of fields may include other documents, arrays, and arrays of documents.

Graph Database

Graph databases are very well suited for problem spaces where we have connected data, such as social networks, spatial data, routing information, and recommendation engines. The following graph shows an example of a social network graph. Given the people (nodes) and their relationships (edges), you can find out who the "friends of friends" of a particular person are—for example, the friends of Howard's friends. For example, geographic map data used in Google Maps is stored in set of relationships or links between points. For intensive data relationship handling, graph databases improve performance by several orders of magnitude. Tech giants like Google, Facebook, and LinkedIn use graph databases to deliver scalable, insightful, and quick service

Neo4j is a highly scalable and most popular ACID-compliant transactional database with native graph storage and processing. It is an open-source graph database, implemented in Java, and accessible from software written in other languages.The first popular NOSQL database was HBase, which is a part of the Hadoop family. The most popular NOSQL database used today is Apache Cassandra, which was developed and owned by Facebook till it was released as open source in 2008. Other NOSQL database systems are SimpleDB, Google'sBigTable, MemcacheDB, Oracle NOSQL, Voldemort, etc.

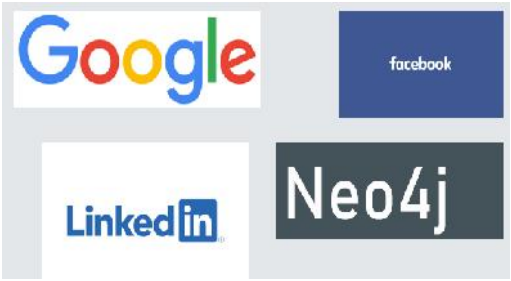


Figure 8: Graph Database

4.4 Data Model

A data model is the model through which we perceive and manipulate our data. For people using a database, the data model describes how we interact with the data in the database. This is distinct from a storage model, which described how the database stores and manipulates the data internally. In an ideal world, we should be ignorant of the storage model, but in practice we need at least some in linking of it- primarily to achieve decent performance.

Database	Storage Model
How we interact with the data in the database	How the database stores and manipulates the data internally

Figure 9: Database is distinct from storage model

Data Models: Example

In conversation, the term "data model" often means the model of the specific data in an application. A developer might point to an entity-relationship diagram of their database and refer to that as their data model containing customers, orders, products, and the like.

What is the Name of Data Model from Last Couple of Decades?

The dominant data model of the last couple of decades is the relational data model, which is best visualized as a set of tables, rather like a page of spread-sheet. Each table has rows, with each row representing some entity of interest. We describe this entity through columns, each having a single value. A column may refer to another row in the same or different table, which constitutes a relationship between those entities.

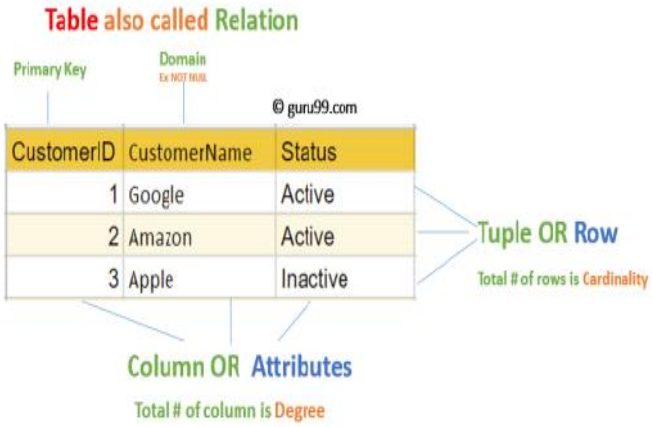


Figure 10: Relational data model

4.5 Introduction to NOSQL

One of the most obvious shifts with NOSQL is a move away from the relational model. Each NOSQL solution has a different model that it uses, which we put into four categories widely used in the NOSQL ecosystem:

Key-value,

Document,

Column-family

And graph

Of course, the first three share a common characteristic of their data models which we will call a aggregate orientation.

4.6 Relational Model vs Aggregate Data Model

Relational Data Model

The relational model takes the information that we want to store and divides it into tuples(rows). A tuple is limited data structure. It captures a set of values, so you cannot nest one tuple within another to get nested records, nor can you put a list of values or tuples within another. This simplicity underpins the relational model-it allows us to think of all operations as operating and returning tuples. This simplicity characterizes the relational model. It allows us to think on data manipulation as operation that have: – As input tuples, and –Return tuples • Aggregate orientation takes a different approach.

Aggregate Data Model

Relational database modeling is vastly different than the types of data structures that application developers use. An aggregate is a collection of data that we interact with as a unit. These units of data or aggregates form the boundaries for ACID operations with the database, Key-value, Document, and Column-family databases can all be seen as forms of aggregate-oriented database.

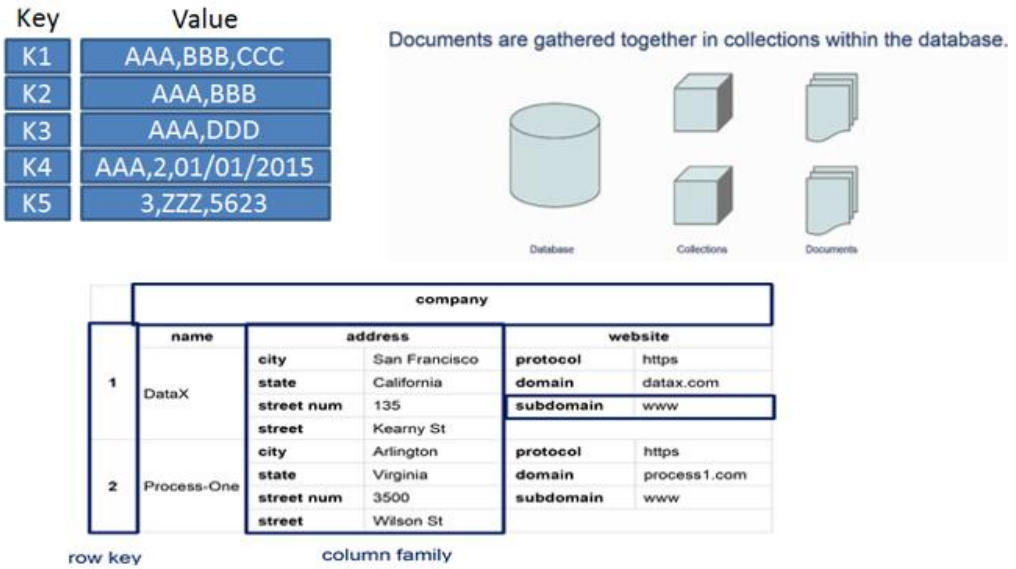


Figure 11: Aggregate data model

Aggregates make it easier for the database to manage data storage over clusters, since the unit of data now could reside on any machine and when retrieved from the database gets all the related data along with it. Aggregate-oriented databases work best when most data interaction is done with the same aggregate, for example when there is need to get an order and all its details, it better to store order as an aggregate object but dealing with these aggregates to get item details on all the orders is not elegant.

Examples of Relations and Aggregates

We have to build and e-commerce website; we are going to be selling items directly to customers over the web, and we will have to store information about users, our product catalogue, orders, shipping addresses, billing addresses and a payment data.

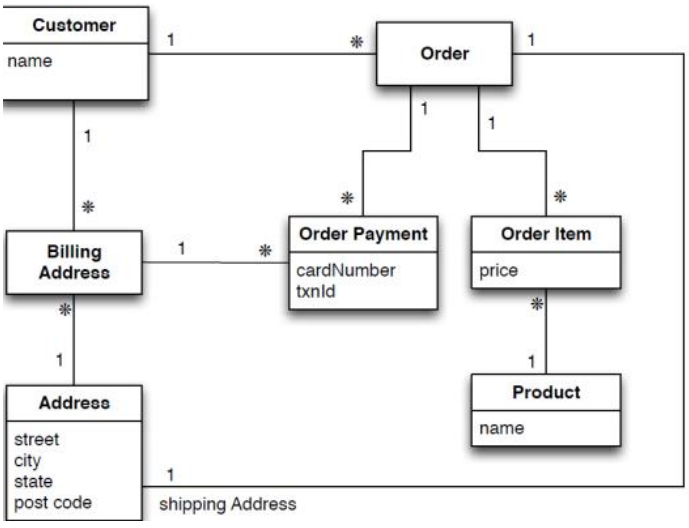


Figure 12: Data Model Oriented Around a Relational Database

We can use this scenario to model the data using a relation data store as well as NOSQL data stores and talk about their pros and cons.. For relational model, we start with a data model shown in this figure. As we are good relational soldiers, everything is properly normalized, so that no data is repeated in multiple tables. We also have referential integrity.

A realistic order system would naturally be more involved than this, but this is the benefit of the rarefied air of a book.Let's see how this model looks when we think in more aggregate-oriented terms:

In this model, we have two main aggregates customer an order. We have used the black-diamond composition marker in UMLto show how data fits into the aggregation structure. The customer contains a list of billing addresses; the order contains a list of order items,a shipping address and payments, The payment itself contains a billing address for the payment. A single logical address record appears three times in the example data, but instead of using IDs its treated as a value and copied each time. This fits the domain where we would not want the shipping address, nor the payments billing address, itchanges. In relational database, we would ensure that the address rows are not updated for this case, making newrow instead. With aggregates, we can copy the whole address structure into the aggregate as we need to. The link between the customer and the order isn't within either aggregate-it's a relationship between aggregates. Similarly, the link form an order item would cross into a separate aggregate structure for products, which we have notgone into. We have shown the product name as part of the order item here-this kind of denormalization is similar to the trade-offs with relational databases, but is more common with aggregates because we want to minimize the number foraggregates we access during a data interaction.

Example of Aggregate Model

The link between the customer and the order is a relationship between aggregates.

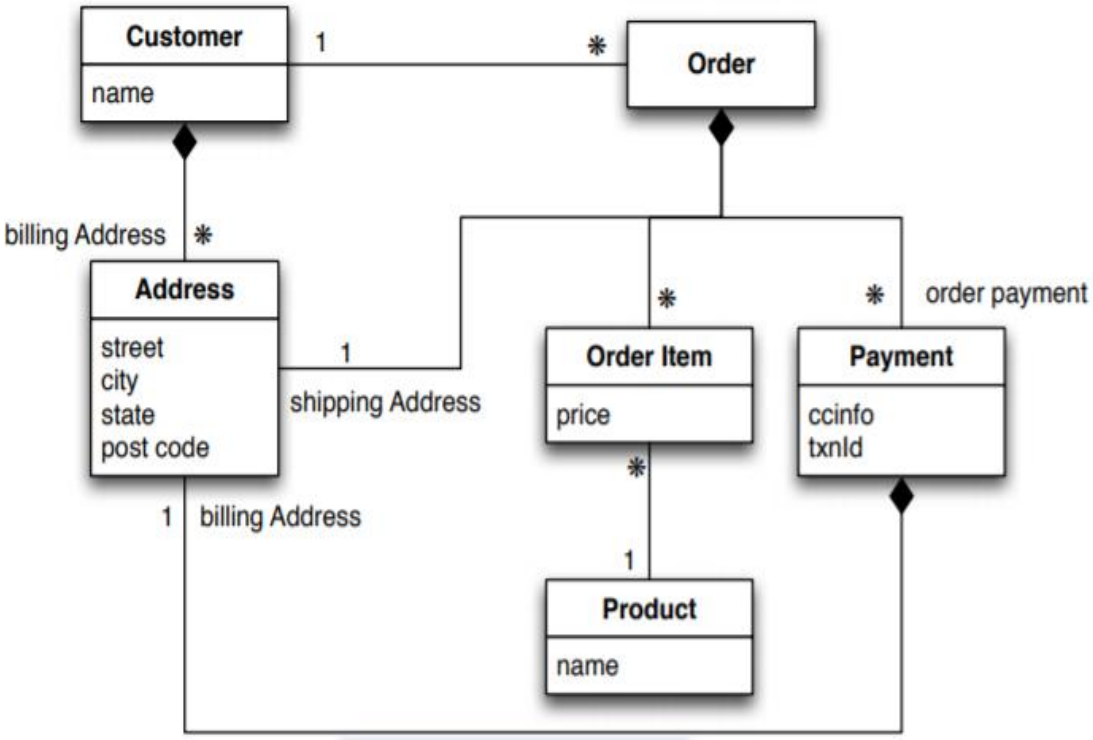


Figure 13: Example of Aggregate data model

```
//Customer
 {
       "id": 1,
       "name": "Fabio",
      "billingAddress": [
       {
            "city": "Bari"
       }
    ]
}
//Orders
{
        "id": 99,
        "customerId": 1,
        "orderItems": [
       {
           "productId": 27,
            "price": 34,
            "productName": "NoSQL Distilled"
       } ],
"shippingAddress": [
 {
      "city": "Bari"} ],
"orderPayment": [
{ "ccinfo": "100-432423-545-134",
  "txnId": "afdfsdfsd",
 "billingAddress": [ {"city": "Chicago" }]
} ]
}
```

important thing to notice here isn't the particular way we have drawn the aggregate boundary so much as the fact that you have to think about accessing that data- and make that part of your thinking when developing the application data model. Indeed, we could draw aggregate boundaries differently, putting all the orders for a customer into the customer aggregate.

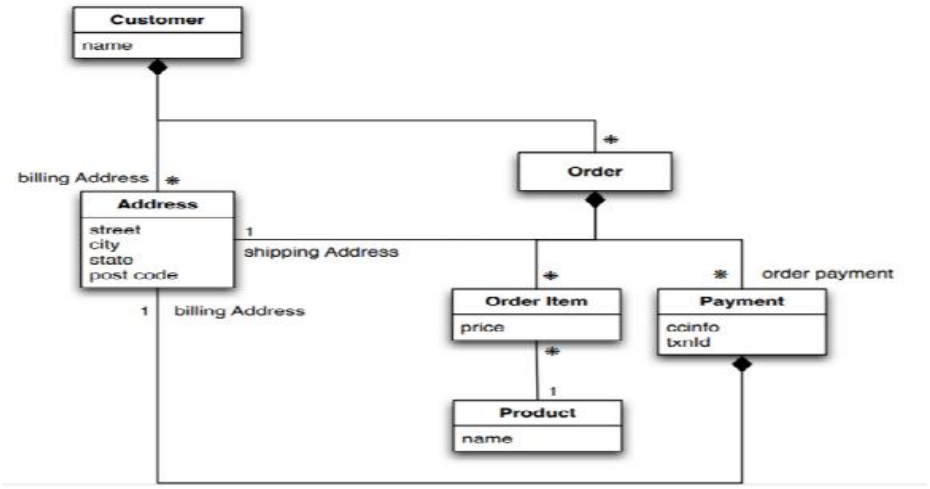


Figure 14: Embed all the objects for customer and the customer's order

Orders { "id": 99, "customerId": 1, "orderItems": [{ "productId": 27, "price": 34, "productName": "NoSQL Distilled"}], "shippingAddress": [{"city": "Chicago"}], "orderPayment": [{ "ccinfo": "100-432423-545-134", "txnId": "afdfsdfsd", "billingAddress": [{"city": "Chicago" }] }] }	• There is the customer id • The product name is a part of the ordered Items. • The product id is part of the ordered items. • The address is stored several times.

Like most things in modelling, there's no universal answer for how to draw your aggregate boundaries. It depends entirely in how you tend to manipulate your data. If you tend to access a customer together with all of that customer's order at once, then you would prefer a single aggregate. However, if you tend to focus on accessing a single order at a time, then you should prefer having separate aggregates for each other. Naturally, this is very context-specific; some applications will prefer one or the other, even within a single system, which is exactly why many people prefer aggregate ignorance.

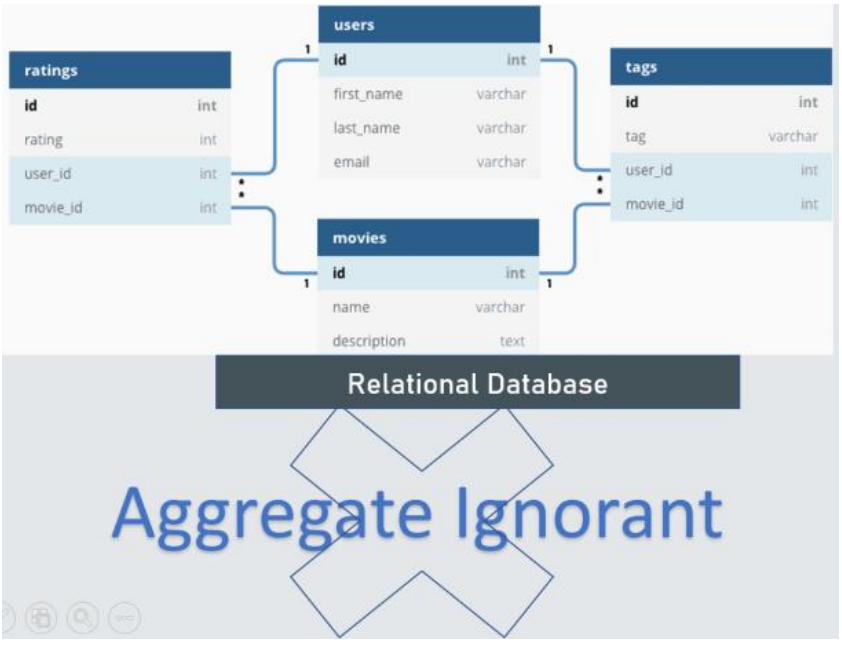


Figure 15: Aggregate Ignorant

Consequences of Aggregate Models

Relational mapping captures various data elements and their relationships. Order consists of

- Order items, a shipping address, and a payment

All can be expresses in the relational model in terms of foreign key relationships. While the relation Mapping captures the various data elements and their relationships well, it does so without any notion of an aggregate entity. In our domain language, we might say that an order consists of order items, a shipping address, and a payment. This can be expressed in the relational model in terms of foreign key relationships-but there is nothing distinguish relationships that represents aggregations from those that don't. As a result, the database can't use a knowledge of aggregate structure to help it store and distribute the data.

The database can't use a knowledge of aggregate structure to help it store and distribute the data. Various data modeling techniques have provided ways of marking aggregate or composite structures. The problem, however is that modelers rarely provide any semantics for what make an aggregate relationship different from any other; where there are semantics, they vary. When working with aggregate-oriented databases, we have a clearer semantics to consider by focusing on the unit of interaction with the data storage. It is however, not alogicaldataproperty: It's all about how the data is being used by applications-a concern that is often outside the bounds of data modeling. Relational databases have no concept of aggregate within their data model, so we call them aggregate-ignorant. In the NOSQL world, graph databases are also aggregate-ignorant. In the NOSQL world, graph databases are also aggregate-ignorant. Being aggregate-ignorant is not a bad thing. It's often difficult to draw aggregate boundaries well, particularly if the same data is used in many different contexts. An order makes a good aggregate when a customer is making and reviewing orders, and when the retailer is processing orders. However, if a retailer wants to analyze its product sales over the last few months, then an order aggregate becomes a trouble. To get to product sales history, you will have to dig into every aggregate in the database. So, an aggregate structure may help with some data interactions but be an obstacle for others.An aggregate-ignorant model allows you to easily look at the data in different ways, so it is a better choice when you don't have aprimary structure for manipulating your data.

Aggregate and Operations

An order is a good aggregate when: A customer is making and reviewing an order, and – When the retailer is processing orders • However, when the retailer want to analyze its product sales over the last months, then aggregate are trouble. • We need to analyze each aggregate to extract sales history.• An order is a good aggregate when: A customer is making and reviewing an order, and – When the retailer is processing orders • However, when the retailer want to analyze its product sales over the last months, then aggregate are trouble. • We need to analyze each aggregate to extract sales history.Aggregate may help in some operation and not in others. In cases where there is not a clear view aggregate ignorant database are the best option. • But, remember the point that drove us to aggregate models (cluster distribution). Running databases on a cluster is need when dealing with huge quantities of data.

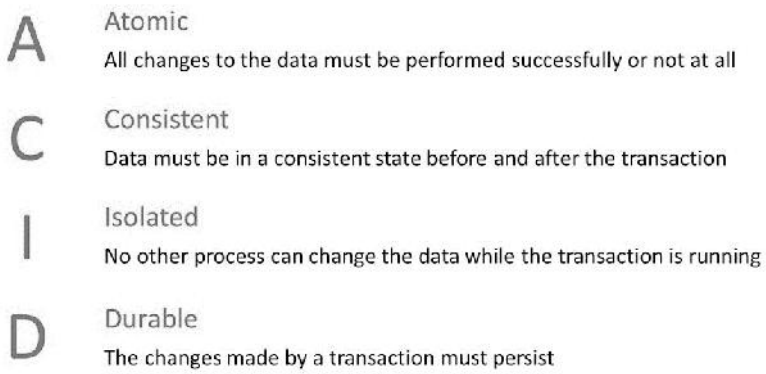
What is the Clinching Reason for Aggregate Orientation?

The clinching reason for aggregate orientation is that it helps greatly with running on a cluster, which as you will remember is the killer argument for the rise of NOSQL. If we are running on a cluster, we need to minimize how many nodes we need to query when we are gathering data. By explicitly including aggregates, we give the database important information about which bits of data will be manipulated together, and thus should live on the same node.

Running on a Cluster

It gives several advantages on computation power and data distribution. However, it requires minimizing the number of nodes to query when gathering data.By explicitly including aggregates, we give the database an important of which information should be stored together.

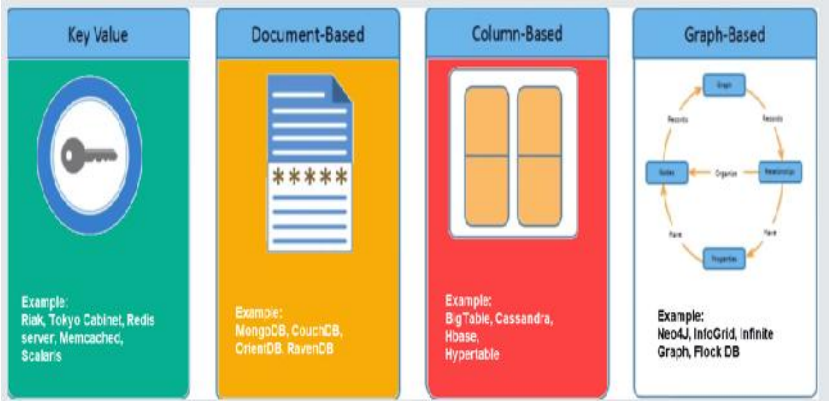
How Aggregates Plays an Important Consequence for Transactions?



NOSQL databases are capable of storing and processing big data which is characterized by various properties such as volume, variety and velocity. Such databases are used in a variety of user applications that need large volume of data which is highly available and efficiently accessible. But they do not enforce or require strong data consistency nor do they support transactions. For example, social media such as Twitter and Facebook [5] generate terabytes of daily data which is beyond the processing capabilities of relational databases. Such applications need high performance but may not need strong consistency. Different vendors design and implement NOSQL databases differently. Indeed, there are different types of NOSQL databases such as document databases, key-value databases, column stores and graph databases. But their common objective is to use data replication in order to ensure high efficiency, availability and scalability of data.

Types of NOSQL Databases

- Majority of NOSQL databases support eventual consistency instead of strong consistency. They do not support database transactions which ensure strong data consistency. Eventual consistency guarantees that all updates will reach all replicas after a certain delay. It works for certain applications such as social media, advertisement records, etc. But some of the user's applications need strong data consistency. That's, bank account data must be consistent whenever any updates are made to data. In other applications, such as online multiplayer gaming applications usually store profile data of a large number of users that require strong consistency. Therefore, NOSQL databases would be useful for managing data in such applications.Bank account data must be consistent whenever any updates are made to data.Online multiplayer gaming applications usually store profile data of a large number of users that require strong consistency.However, the lack of support for transactions, table joins and referential integrity in NOSQL databases, mean that they are not suitable for applications such as banking, online gaming, etc. Such applications require all data replicas should be made instantly consistent and applications should have the latest version of the data in the case of any updates.



4.7 Introduction to NOSQL

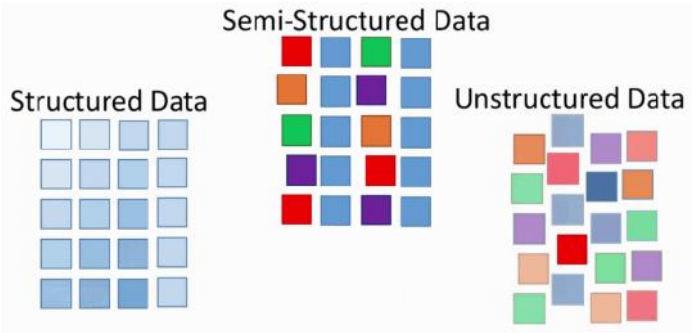
Such applications require all data replicas should be made instantly consistent and applications should have the latest version of the data in the case of any updates.

Figure 16: Non-relational database management system

What is the Major Purpose of Using a NOSQL Database?

The major purpose of using a NOSQL database is for distributed data stores with huge data storage needs. NOSQL is used for Big data and real-time web apps.

The major purpose of using a NOSQL database is for distributed data stores with humongous data storage needs. NOSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data. **NOSQL database** stands for "Not Only SQL" or "Not SQL." Though a better term would be "NOREL", NOSQL caught on. Carlo Strozzi introduced the NOSQL concept in 1998.Traditional RDBMS uses SQL syntax to store and retrieve data for further insights. Instead, a NOSQL database system encompasses a wide range of database technologies that can store structured, semi-structured, unstructured and polymorphic data.



Example of SQL Database

Relational databases are **table-based**. NOSQL databases can be document based, graph databases, key-value pairs, or wide-column stores. Relational databases were built during a time that data was mostly structured and clearly defined by their relationships. Today, we know that data today is much more complex.

Why NOSQL?

The concept of NOSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data. The system response time becomes slow when you use RDBMS for massive volumes of data.To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive. The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."NOSQL database is non-relational, so it scales out better than relational databases as they are designed with web applications in mind.

History of NOSQL Databases

- 1998- Carlo Strozzi use the term NOSQL for his lightweight, open-source relational database
- 2000- Graph database Neo4j is launched
- 2004- Google BigTable is launched
- 2005- CouchDB is launched
- 2007- The research paper on Amazon Dynamo is released
- 2008- Facebooks open sources the Cassandra project
- 2009- The term NOSQL was reintroduced

Features of NOSQL

- Non-relational
- Never follow the relational model
- Never provide tables with flat fixed-column records
- Work with self-contained aggregates or BLOBs
- Doesn't require object-relational mapping and data normalization
- No complex features like query languages, query planners, referential integrity joins, ACID
- Schema-free

NOSQL databases are either schema-free or have relaxed schemas. Do not require any sort of definition of the schema of the data. Offers heterogeneous structures of data in the same domain

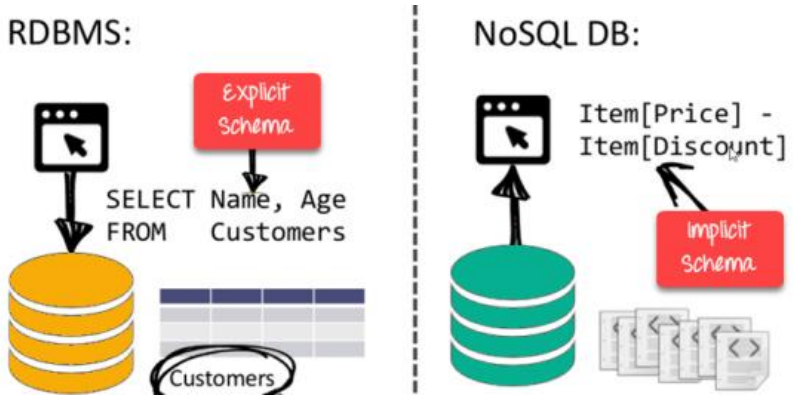


Figure 17: Schema Free

- Simple API

Offers easy to use interfaces for storage and querying data provided. APIs allow low-level data manipulation & selection methods. Text-based protocols mostly used with HTTP REST with JSON. Mostly used no standard based NOSQL query language. Web-enabled databases running as internet-facing services

- Distributed
  - Multiple NOSQL databases can be executed in a distributed fashion. Offers auto-scaling and fail-over capabilities. Often ACID concept can be sacrificed for scalability and throughput.
  - Mostly no synchronous replication between distributed nodes Asynchronous Multi-Master Replication, peer-to-peer, HDFS Replication. Only providing eventual consistency

- Shared Nothing Architecture. This enables less coordination and higher distribution.

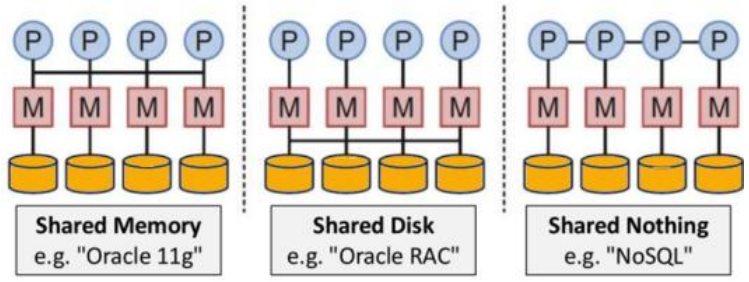
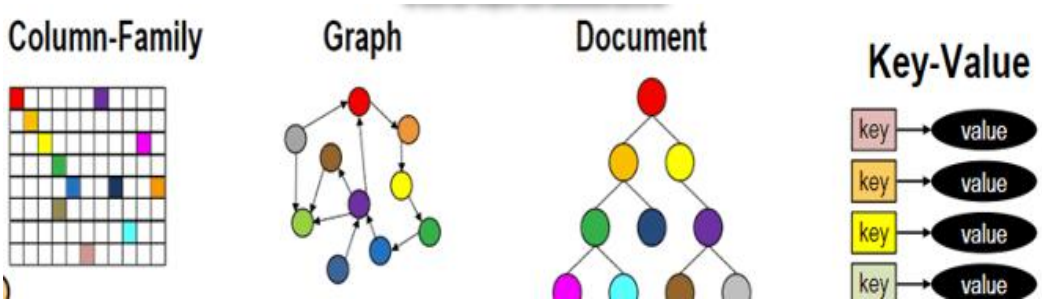


Figure 18: NoSQL is shared nothing

4.8 Types of NOSQL Databases

NOSQL Databases are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented. Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.



Key Value Pair Based

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load. Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.A key-value (an associative array, map, symbol table,or dictionary**)** is an abstract data type composed of a collection of key/value pairs, such that each possible key appears just once in the collection.

- "Pride and Prejudice": "Alice",
- "The Brothers Karamazov": "Pat",
- "Wuthering Heights": "Alice"

It is one of the most basic NOSQL database examples. This kind of NOSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.Redis, Dynamo, Riak are some NOSQL examples of key-value store DataBases. They are all based on Amazon's Dynamo paper.

Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

Column-Based

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column. Column-based NOSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs,HBase, Cassandra, HBase, Hyper table are NOSQL query examples of column based database.A column of a distributed data store is a NOSQL object of the lowest level in a keyspace. It is a tuple (a key-value pair) consisting of three elements:

Unique name: Used to reference the column

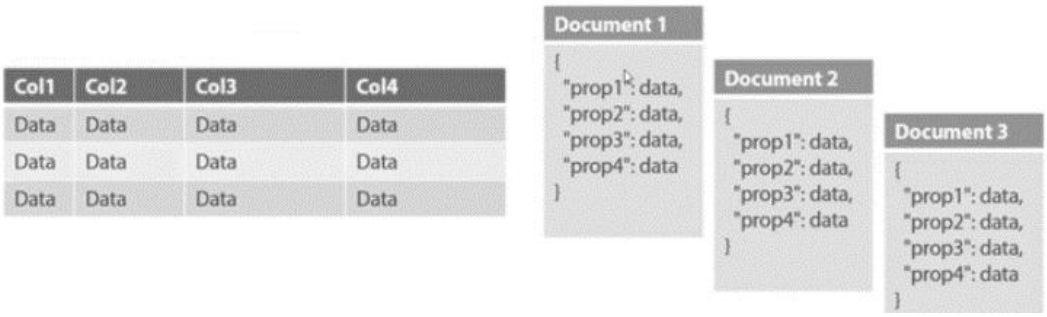
Value: The content of the column. It can have different types, like AsciiType, LongType, TimeUUIDType, UTF8Type among others.

Timestamp: The system timestamp used to determine the valid content.

```
{
   street: {name: "street", value: "1234 x street", timestamp: 123456789},
   city: {name: "city", value: "sanfrancisco", timestamp: 123456789},
   zip: {name: "zip", value: "94107", timestamp: 123456789},
}
```

Document-Oriented

In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON.



Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems

Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge have a unique identifier.Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them. Graph base database mostly used for social networks, logistics, spatial data.

Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.A graph database is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. A graph database is any storage system that provides index-free adjacency. This means that every element contains a direct pointer to its adjacent elements and no index lookups are necessary. General graph databases that can store any graph are distinct from specialized graph databases such as triplestores and network databases.

Query Mechanism Tools for NOSQL

- The most common data retrieval mechanism is the REST-based retrieval of a value based on its key/ID with GET resource. Document store Database offers more difficult queries as they understand the value in a key-value pair.
- For example, CouchDB allows defining views with MapReduce

4.9 Introduction to Distribution Models

The primary driver of interest in NOSQLhas been its ability to run databases on a large cluster. As data volumes increase, it becomes more difficult and expensive to scale up—buy a bigger server to run the database on. A more appealing option is to scale out—run the database on a cluster of servers. Aggregate orientation fits well with scaling out because the aggregate is a natural unit to use for distribution.

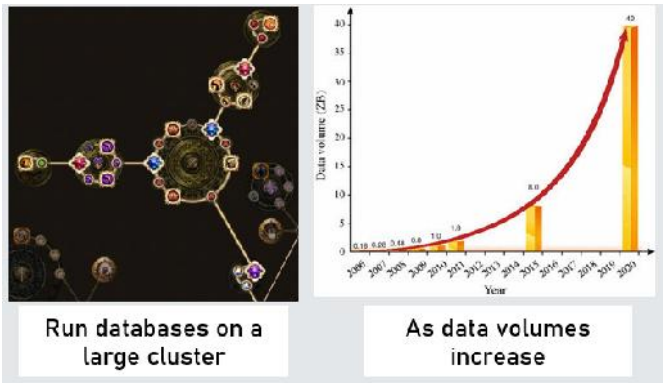


Figure 19: Distribution model

Depending on your distribution model, you can get a data store that will give you the ability to handle larger quantities of data, the ability to process a greater read or write traffic, or more availability in the face of network slowdowns or breakages. These are often important benefits, but they come at a cost. Running over a cluster introduces complexity—so it's not something to do unless the benefits are compelling.

Important Benefits of Distribution Model

Depending on your distribution model, you can get a data store that will give you the ability to handle larger quantities of data, the ability to process a greater read or write traffic, or more availability in the face of network slowdowns or breakages.

Figure 20: Handle large quantities of data

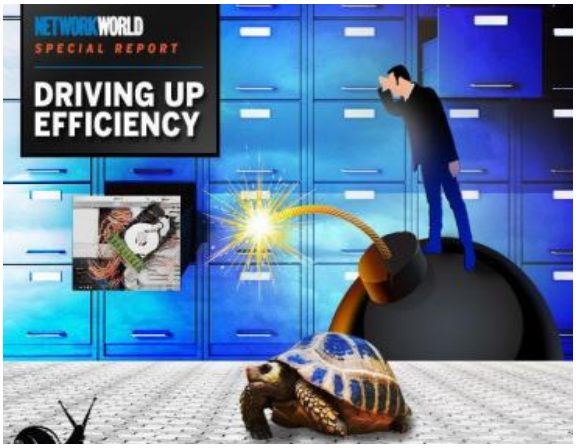


Figure 21: Network slowdown or breakages

These are often important benefits, but they come at a cost. Running over a cluster introduces complexity—so it's not something to do unless the benefits are compelling.Broadly, there are two paths to data distribution: replication and sharding.

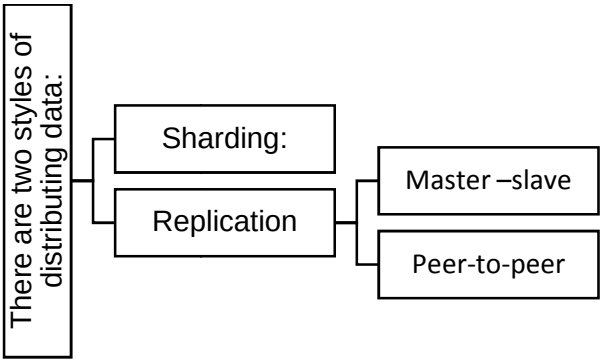


Figure 22: Two styles of distributing data

Replication takes the same data and copies it over multiple nodes. Sharding puts different data on different nodes.

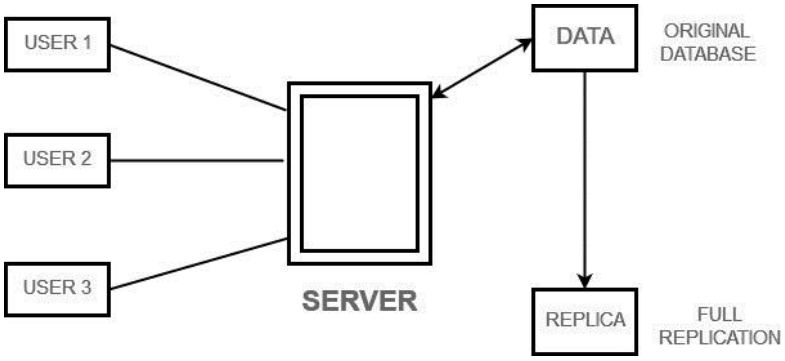


Figure 23: Replication

Sharding puts different data on different nodes. Replication and sharding are orthogonal techniques: You can use either or both of them.

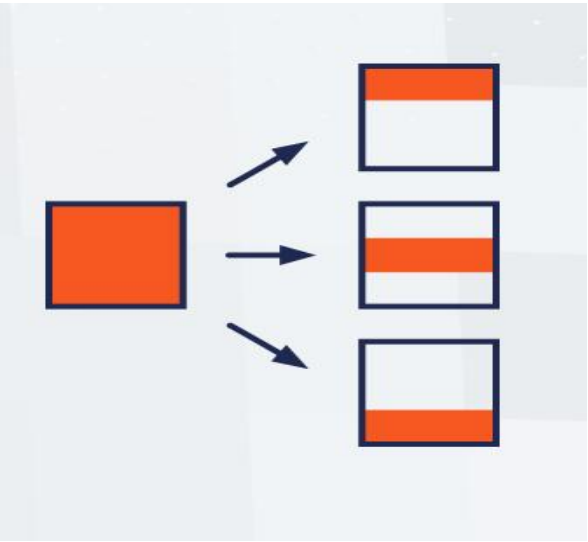


Figure 24: Sharding

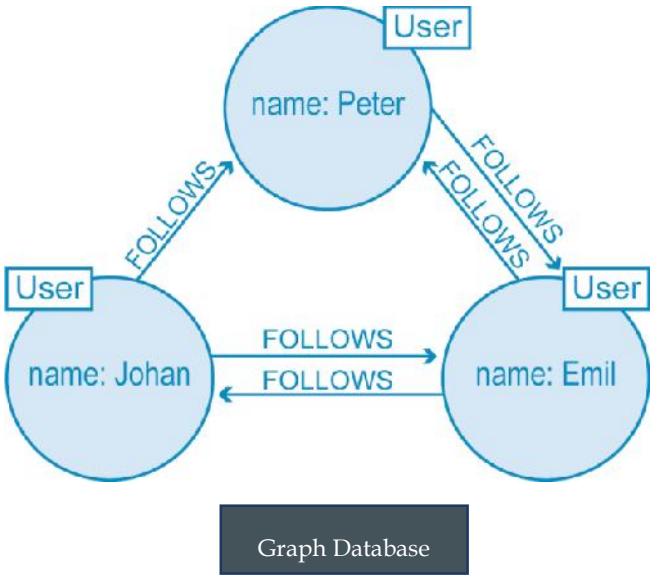
Replication comes into two forms: master-slave and peer-to-peer. We will now discuss these techniques starting at the simplest and working up to the more complex: first single-server, then master-slave replication, then sharding, and finally peer-to-peer replication.

Single Server

The first and the simplest distribution option is the one we would most often recommend—no distribution at all. Run the database on a single machine that handles all the reads and writes to the data store. We prefer this option because it eliminates all the complexities that the other options introduce; it's easy for operations people to manage and easy for application developers to reason about.

Although a lot of NOSQLdatabases are designed around the idea of running on a cluster, it can make sense to use NOSQLwith a single-server distribution model if the data model of the NOSQL store is more suited to the application. Graph databases are the obvious category here—these work best in a single-server configuration

Question: Which Database Works Best in Single-Server Configuration?

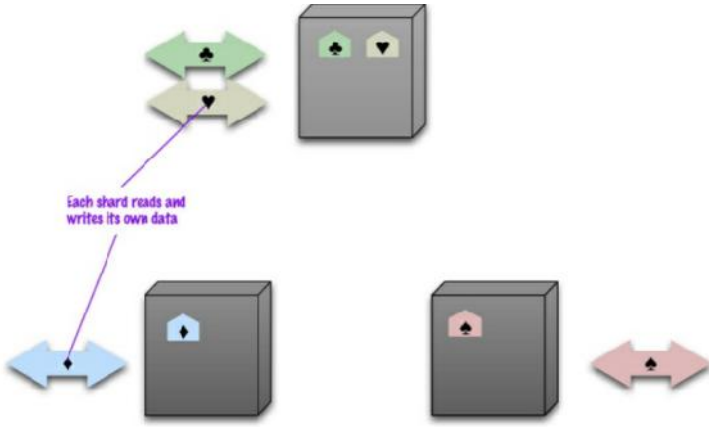


What is the Most Appropriate Solution for Processing Aggregates in a Single-Server Document?

If your data usage is mostly about processing aggregates, then a single-server document or key-value store may well be worthwhile because it's easier on application developers

Sharding

Often, a busy data store is busy because different people are accessing different parts of the dataset. In these circumstances we can support horizontal scalability by putting different parts of the data onto different servers—a technique that's called sharding.



- **Sharding** is the practice of optimizing database management systems by separating the rows or columns of a larger database table into multiple smaller tables. The new tables are called "shards" (or partitions), and each new table either has the same schema but unique rows (as is the case for "horizontal sharding") or has a schema that is a proper subset of the original table's schema (as is the case for "vertical sharding").

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston
3	Carrie	Conway	Chicago
4	David	Doe	Denver

Vertical Shards

VS1

CUSTOMER ID	FIRST NAME	LAST NAME
1	Alice	Anderson
2	Bob	Best
3	Carrie	Conway
4	David	Doe

VS2

CUSTOMER ID	CITY
1	Austin
2	Boston
3	Chicago
4	Denver

Horizontal Shards

HS1

CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston

HS2

CUSTOMER ID	FIRST NAME	LAST NAME	CITY
3	Carrie	Conway	Chicago
4	David	Doe	Denver

Figure 25: Vertical and horizontal shards

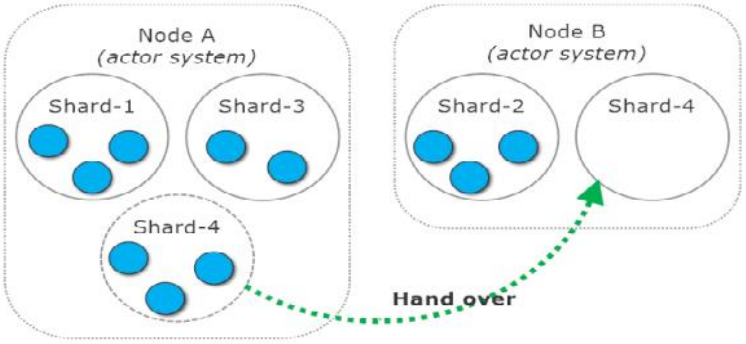
Why Is Sharding Used?

- In the ideal case, we have different users all talking to different server nodes. Each user only has to talk to one server, so gets rapid responses from that server. The load is balanced out nicely between servers. By sharding a larger table, you can store the new chunks of data, called logical shards, across multiple nodes to achieve horizontal scalability and improved performance. Once the logical shard is stored on another node, it is referred to as a physical shard. With massively parallel processing, you can take advantage of all the compute resources across your cluster for every query. Because the

individual shards are smaller than the logical table as a whole, each machine has to scan fewer rows when responding to a query.for example, if we have ten servers, each one only has to handle 10% of the load. Of course the ideal case is a pretty rare beast. In order to get close to it we have to ensure that data that's accessed together is clumped together on the same node and that these clumps are arranged on the nodes to provide the best data access.

How to Clump the Data up so that One User Mostly gets her Data from a Single Server?

- The first part of this question is how to clump the data up so that one user mostly gets her data from a single server. This is where aggregate orientation comes in really handy. The whole point of aggregates is that we design them to combine data that's commonly accessed together—so aggregates leap out as an obvious unit of distribution. When it comes to arranging the data on the nodes, there are several factors that can help improve performance. If you know that most accesses of certain aggregates are based on a physical location, you can place the data close to where it's being accessed. If you have orders for someone who lives in Boston, you can place that data in your eastern US data centre. Another factor is trying to keep the load even. This means that you should try to arrange aggregates so they are evenly distributed across the nodes which all get equal amounts of the load. This may vary over time, for example if some data tends to be accessed on certain days of the week—so there may be domain-specific rules you'd like to use.In some cases, it's useful to put aggregates together if you think they may be read in sequence. The Bigtable paper [Chang etc.] described keeping its rows in lexicographic order and sorting web addresses based on reversed domain names (e.g., com.martinfowler). This way data for multiple pages could be accessed together to improve processing efficiency.
- Historically most people have done sharding as part of application logic. You might put all customers with surnames starting from A to D on one shard and E to G on another. This complicates the programming model, as application code needs to ensure that queries are distributed across the various shards.
- Furthermore, rebalancing the sharding means changing the application code and migrating the data. Many NOSQLdatabases offer auto-sharding, where the database takes on the responsibility of allocating data to shards and ensuring that data access goes to the right shard. This can make it much easier to use sharding in an application.



Rebalancing the Sharding

When Horizontal Sharding is Effective?

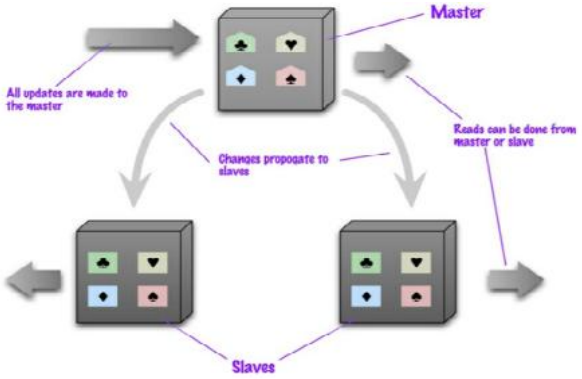
Horizontal sharding is effective when queries tend to return a subset of rows that are often grouped together. For example, queries that filter data based on short date ranges are ideal for horizontal sharding since the date range will necessarily limit querying to only a subset of the servers.

When Vertical Sharding is Effective?

Vertical sharding is effective when queries tend to return only a subset of columns of the data. For example, if some queries request only names, and others request only mail-id, then the names and Mail-id can be sharded onto separate servers.

Sharding is particularly valuable for performance because it can improve both read and write performance. Using replication, particularly with caching, can greatly improve read performance but does little for applications that have a lot of writes.

Sharding provides a way to horizontally scale writes. Sharding does little to improve resilience when used alone. Although the data is on different nodes, a node failure makes that shard's data unavailable just as surely as it does for a single-server solution. The resilience benefit it does provide is that only the users of the data on that shard will suffer; however, it's not good to have a database with part of its data missing. With a single server it's easier to pay the effort and cost to keep that server up and running; clusters usually try to use less reliable machines, and you're more likely to get a node failure. So in practice, sharding alone is likely to decrease resilience

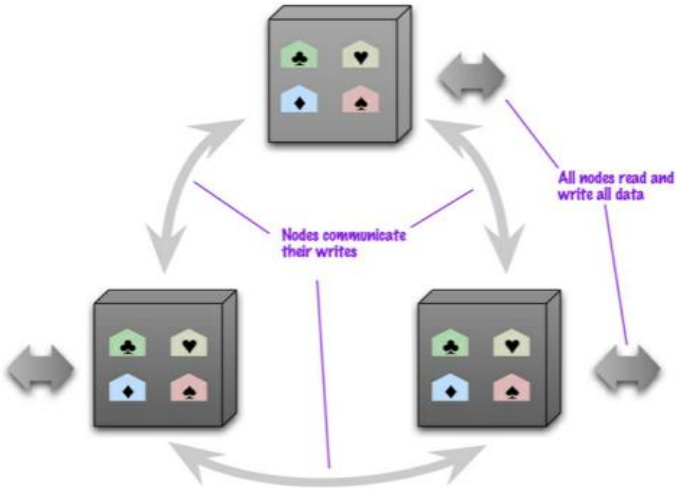


Master-Slave Replication

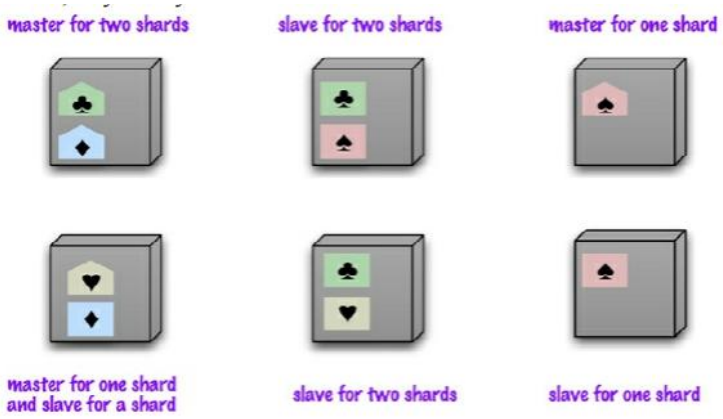
With master-slave distribution, you replicate data across multiple nodes. One node is designated as the master, or primary. This master is the authoritative source for the data and is usually responsible for processing any updates to that data. The other nodes are slaves, or secondaries. A replication process synchronizes the slaves with the master (see Figure 4.2).Master-slave replication is most helpful for scaling when you have a read-intensive dataset. You can scale horizontally to handle more read requests by adding more slave nodes and ensuring that all read requests are routed to the slaves. You are still, however, limited by the ability of the master to process updates and its ability to pass those updates on. Consequently, it isn't such a good scheme for datasets with heavy write traffic, although offloading the read traffic will help a bit with handling the write load.A second advantage of master-slave replication is read resilience: Should the master fail, the slaves can still handle read requests. Again, this is useful if most of your data access is reads. The failure of the master does eliminate the ability to handle writes until either the master is restored or a new master is appointed. However, having slaves as replicates of the master does speed up recovery after a failure of the master since a slave can be appointed a new master very quickly

Peer-to-Peer Replication

Master-slave replication helps with read scalability but doesn't help with scalability of writes. It provides resilience against failure of a slave, but not of a master. Essentially, the master is still a bottleneck and a single point of failure. Peer-to-peer replication (see Figure 4.3) attacks these problems by not having a master. All the replicas have equal weight, they can all accept writes, and the loss of any of them doesn't prevent access to the data store. The prospect here looks mighty fine. With a peer-to-peer replication cluster, you can ride over node failures without losing access to data. Furthermore, you can easily add nodes to improve your performance. There's much to like here—but there are complications. The biggest complication is, again, consistency. When you can write to two different places, you run the risk that two people will attempt to update the same record at the same time—a write-write conflict. Inconsistencies on read lead to problems but at least they are relatively transient. Inconsistent writes are forever



Combining Sharding and Replication



Replication and sharding are strategies that can be combined. If we use both master-slave replication and sharding (see Figure 4.4), this means that we have multiple masters, but each data item only has a single master. Depending on your configuration, you may choose a node to be a master for some data and slaves for others, or you may dedicate nodes for master or slave duties.Using peer-to-peer replication and sharding is a common strategy for column-family databases. In a scenario like this you might have tens or hundreds of nodes in a cluster with data sharded over them. A good starting point for peer-to-peer replication is to have a replication factor of 3, so each shard is present on three nodes. Should a node fail, then the shards on that node will be built on the other nodes (see Figure 4.5).

What is the Difference between Sharding and Partitioning?

Sharding and partitioning are both about breaking up a large data set into smaller subsets. The difference is that sharding implies the data is spread across multiple computers while partitioning does not. Partitioning is about grouping subsets of data within a single database instance. In many cases, the terms sharding and partitioning are even used synonymously, especially when preceded by the terms "horizontal" and "vertical." Thus, "horizontal sharding" and "horizontal partitioning" can mean the same thing.

4.10 Introduction to Hadoop Partitioner

- Partitioner allows distributing how outputs from the map stage are send to the reducers. Partitioner controls the keys partition of the intermediate map-outputs. The key or a subset of the key is used to derive the partition by a hash function. With the help of hash function, key (or a subset of the key) derives the partition. The total number of partitions is

equal to the number of reduce tasks.Partitioner in MapReduce job execution controls the partitioning of the keys of the intermediate map-outputs. With the help of hash function, key (or a subset of the key) derives the partition. The total number of partitions is equal to the number of reduce tasks.Partitions runs on the same machine where the mapper had completed its execution by the consuming the mapper output. Entire mapper output sent to partitioner.

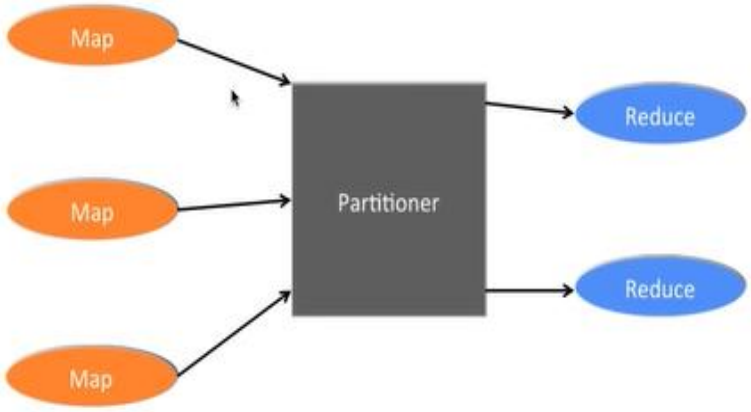


Figure 26: Hadoop Partitioner

What is Hadoop Partitioner?

- Controls the partitioning of the keys
- Keys derive the partition
- Total number of partitions == Number of reduce tasks
- It runs on the same machine
- Entire mapper output sent to partitioner.
- Partitioner forms number of reduce task groups from the mapper output.
- By default. Hadoop framework is hash based partitioner. Hash partitioner partitions the key space by using the hash code.

Partition class decides which partition a given (key, value) pair will go. Partition phase in MapReduce data flow takes place after map phase and before reduce phase.

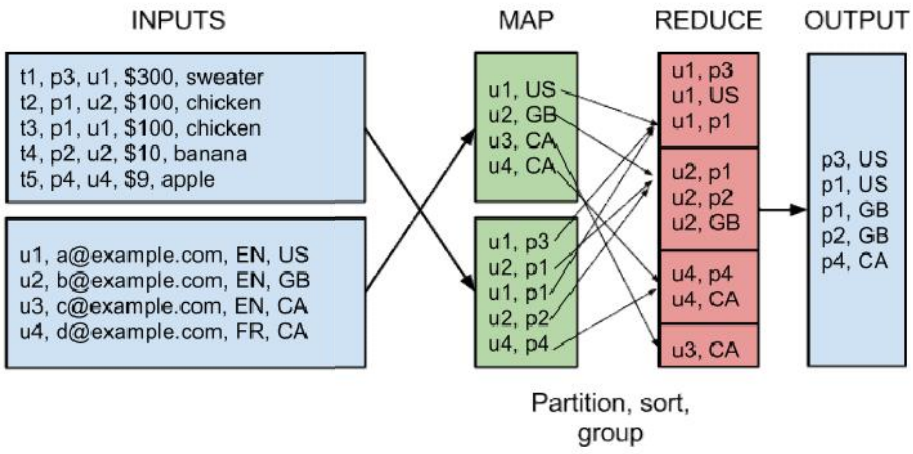


Figure 27: Partitioner, sort and group

Need of MapReduce Partitioner in Hadoop

In MapReduce job execution, it takes an input data set and produces the list of key value pair. These key-value pair is the result of map phase. In which input data are split and each task processes the split and each map, output the list of key value pairs

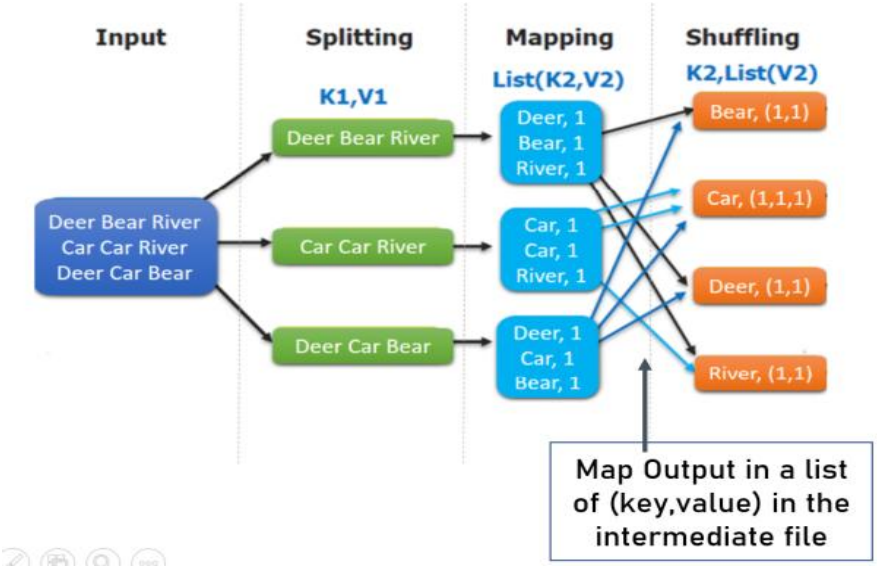


Figure 28: Need of MapReduce Partitioner in Hadoop

- Then, framework sends the map output to reduce task. Reduce processes the user-defined reduce function on map outputs. Before reduce phase, partitioning of the map output take place on the basis of the key.Hadoop Partitioning specifies that all the values for each key are grouped together. It also makes sure that all the values of a single key go to the same reducer. This allows even distribution of the map output over the reducer. Partitioner in a MapReduce job redirects the mapper output to the reducer by determining which reducer handles the particular key.
- Partitioner makes sure that same key goes to the same reducer!

Hadoop Default Partitioner

- **Hash Partitioner** is the default Partitioner. It computes a hash value for the key. It also assigns the partition based on this result.

How many Partitioners in Hadoop?

- The total number of Partitioner depends on the number of reducers. Hadoop Partitioner divides the data according to the number of reducers. It is set by **JobConf.setNumReduceTasks ()** method. Thus the single reducer processes the data from single partitioner. The important thing to notice is that the framework creates partitioner only when there are many reducers.If in data input in MapReduce job one key appears more than any other key. In such case, to send data to the partition we use two mechanisms which are as follows: The key appearing more number of times will be sent to one partition. All the other key will be sent to partitions on the basis of their **hashCode()**.
- If **hashCode()** method does not distribute other key data over the partition range. Then data will not be sent to the reducers.

Poor Partitioning in Hadoop MapReduce

- Poor partitioning of data means that some reducers will have more data input as compared to other. They will have more work to do than other reducers. Thus the entire job has to wait for one reducer to finish its extra-large share of the load.

Summary

- Users may manage preset data relationships across various databases using standard relational databases. Microsoft SQL Server, Oracle Database, MySQL, and IBM DB2 are all examples of typical relational databases.
- Non-tabular databases (sometimes known as "not simply SQL") store data differently from relational tables. NOSQL databases are classified according to their data model. Document, key-value, wide-column, and graph are the most common kinds. They have adaptable schemas and can handle big volumes of data and heavy user loads with ease.
- The software that allows users to update, query, and administer a relational database is known as an RDBMS, or relational database management system. The primary programming language for accessing databases is Structured Query Language (SQL).
- ACIDITY (Atomicity, Consistency, Isolation, Durability) When analysing databases and application architectures, Database Professionals check for ACID (acronym for Atomicity, Consistency, Isolation, and Durability).
- The capacity of a system to increase or decrease in performance and cost in response to changes in application and system processing demands is known as scalability. When considering hardware and software, businesses that are rapidly expanding should pay special attention to scalability.
- In a Hadoop cluster, MapReduce is a programming paradigm that permits tremendous scalability across hundreds or thousands of computers. MapReduce, as the processing component, lies at the heart of Apache Hadoop. The reduction job is always carried out after the map job, as the term MapReduce implies.
- A column-oriented database management system, often known as a columnar database management system, is a database management system that stores data tables by column rather than row. In the realm of relational DBMS, the practical application of a column store vs a row store differs little.
- A graph database is a database that represents and stores data using graph structures for semantic searches, such as nodes, edges, and attributes. The graph is an important notion in the system.

Keywords

Relational Database: A relational database is a collection of data elements that are linked together by pre-defined connections. These elements are laid down in a tabular format with columns and rows. Tables store data about the things that will be represented in the database. A field keeps the actual value of an attribute, while each column in a table carries a specific type of data.

NOSQL Database: Rather than relational tables, NOSQL databases store data as documents. As a result, we categorise them as "not simply SQL" and divide them into several flexible data models. Pure document databases, key-value stores, wide-column databases, and graph databases are examples of NOSQL databases. NOSQL databases are designed from the bottom up to store and handle large volumes of data at scale, and they are increasingly used by modern enterprises.

Relational Database: Relational DataBase Management Systems (RDBMS) is an acronym for Relational DataBase Management Systems. It's an application that lets us build, remove, and update relational databases. A relational database is a database system that stores and retrieves data in the form of rows and columns in a tabular format. It is a minor subset of DBMS that was created in the 1970s by E.F Codd.

Key/Value Store:A key-value store, sometimes known as a key-value database, is a simple database that employs an associative array (think of a map or dictionary) as its basic data model, with each key corresponding to one and only one item in a collection. A key-value pair is the name for this type of connection.

Columnar Database:A column-oriented database management system, often known as a columnar database management system, is a database management system that stores data tables by column rather than row. In the realm of relational DBMS, the practical application of a column store vs a row store differs little.

Graph Database:A graph database is a single-purpose, specialised platform for constructing and managing graphs. Graphs are made up of nodes, edges, and attributes, which are all utilised to represent and store data in a way that relational databases can't.

Aggregate Models: An aggregate is a group of data with which we interact as a whole. The boundaries for ACID operations with the database are formed by these units of data or aggregates. Key-value, Document, and Column-family databases are all examples of aggregate-oriented databases.

Self Assessment

1. A NOSQL database is defined as which of the following?

A. SQLServer

B. MongoDB

C. Cassandra

D. None of the mentioned

2. NOSQL databases is used mainly for handling large volumes of _____ data.

A. Unstructured

B. Structured

C. Semi-structured

D. All of the mentioned

3. NOSQL is useful when an enterprise needs to access, analyze, and utilize massive amounts of either structured or unstructured data

A. Access

B. Analyze

C. Utilize

D. All of the above

4. What is the name of data model from last couple of decades?

A. Relational data model

B. Data mart

C. Supportive data model

D. None of above

5. Table also called a _____

A. Tuple

B. Column

C. Relation

D. None of above

6. What is the clinching reason for aggregate orientation?

A. Greatly with running on a cluster

B. Helps greatly with running on a regression

C. Helps greatly with running on a classification

D. None of the above

7. NOSQL is used for
A. Big data
B. Real-time web apps.
C. Both
D. None of above

8. In which year Carlo Strozzi use the term NOSQL for his lightweight, open-source relational database
A. 1998
B. 2000
C. 2004
D. None of the above

9. Select correct features of NOSQL
A. Non-relational
B. Schema-free
C. Simple API
D. All of the above

10. Which database works best in single-server configuration?
A. Graph database
B. Relational database
C. Both
D. None of above

11. What is sharding?
A. Putting different parts of the data onto central servers
B. Putting different parts of the data onto different servers
C. Both
D. None of above

12. Rebalancing the sharding means _____
A. Changing the application code
B. Migrating the data.
C. Both
D. None of above

13. Which database works best in single-server configuration?
A. Graph database
B. Relational database
C. Both
D. None of above

14. What is sharding?
A. Putting different parts of the data onto central servers

B. Putting different parts of the data onto different servers

C. Both

D. None of above

15. Rebalancing the sharding means _____

A. Changing the application code

B. Migrating the data.

C. Both

D. None of above

Answers for Self Assessment

1.	A	2.	B	3.	D	4.	A	5.	C
6.	A	7.	C	8.	A	9.	D	10.	A
11.	B	12.	C	13.	A	14.	B	15.	C

Review Questions

1. Explain types of NOSQL.
2. Write down features of NOSQL.
3. Write down about data models.
4. Difference between RDBMS vs NOSQL.
5. What is the major purpose of using a NOSQL database.
6. What are the advantages and disadvantages of NOSQL.

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices of Scalable Real time Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NOSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 05: Introduction to Hadoop

CONTENTS

Objectives

- Learn introduction about Hadoop.
- Learn benefits of Hadoop for bigdata
- Learn Open-Source Software Related to Hadoop
- Learn what is big data
- Learn why big data in the cloud makes perfect sense
- Learn Big opportunities, big challenges

Introduction

Hadoop is a framework that allows us to store and process large datasets in parallel and distributed fashion.Two major problems in dealing with BIG DATA

- Storage
- Processing

Storage problem resolved by

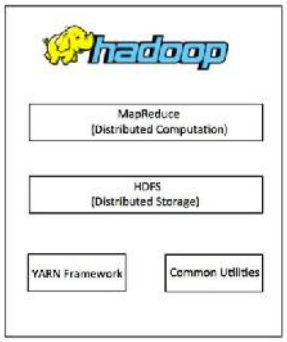
- HDFS

All big amount of data that we are dumping is gets distributed over different machine.These machines are interconnected

Processing problem resolved by

- mapReduce

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models.The Hadoop framework application works in an environment that provides distributed *storage* and *computation* across clusters of computers. . Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.



At its core, Hadoop has two major layers namely −

- Processing/Computation layer (MapReduce), and
- Storage layer (Hadoop Distributed File System)

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications having large datasets.Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules − Hadoop Common − These are Java libraries and utilities required by other Hadoop modules.

Hadoop YARN − this is a framework for job scheduling and cluster resource management. YARN stands for yet another Resource Negotiator. It manages and schedules the resources, and decides what should happen in each data node. The central master node that manages all processing requests is called the Resource Manager. The Resource Manager interacts with Node Managers; every slave datanode has its own Node Manager to execute tasks.

How Hadoop Improves on Traditional Databases

- Capacity: Hadoop stores large volumes of data.

By using a distributed file system called an HDFS (Hadoop Distributed File System), the data is split into chunks and saved across clusters of commodity servers. As these commodity servers are built with simple hardware configurations, these are economical and easily scalable as the data grows.

HDFS is the pillar of Hadoop that maintains the distributed file system. It makes it possible to store and replicate data across multiple servers.HDFS has a NameNode and DataNode. DataNodes are the commodity servers where the data is actually stored. The NameNode, on the other hand, contains metadata with information on the data stored in the different nodes. The application only interacts with the NameNode, which communicates with data nodes as required.

- Speed: Hadoop stores and retrieves data faster.

Hadoop uses the MapReduce functional programming model to perform parallel processing across data sets. So, when a query is sent to the database, instead of handling data sequentially, tasks are

split and concurrently run across distributed servers. Finally, the output of all tasks is collated and sent back to the application, drastically improving the processing speed.

Why is Hadoop Important?

Ability to store and process huge amounts of any kind of data, quickly.

With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.

- **Computing Power**

Hadoop's distributed computing model processes big data fast. The more computing nodes you use, the more processing power you have.

- **Fault Tolerance.**

Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.

- **Flexibility.**

Unlike traditional relational databases, you don’t have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.

- **Low Cost.**

The open-source framework is free and uses commodity hardware to store large quantities of data.

- **Scalability.**

You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

What are the Challenges of Using Hadoop?

- **MapReduce programming is not a good match for all problems.** It’s good for simple information requests and problems that can be divided into independent units, but it's not efficient for iterative and interactive analytic tasks. MapReduce is file-intensive. Because the nodes don’t intercommunicate except through sorts and shuffles, iterative algorithms require multiple map-shuffle/sort-reduce phases to complete. This creates multiple files between MapReduce phases and is inefficient for advanced analytic computing.
- **There’s a widely acknowledged talent gap.** It can be difficult to find entry-level programmers who have sufficient Java skills to be productive with MapReduce. That's one reason distribution providers are racing to put relational (SQL) technology on top of Hadoop. It is much easier to find programmers with SQL skills than MapReduce skills and, Hadoop administration seems part art and part science, requiring low-level knowledge of operating systems, hardware and Hadoop kernel settings.
- **Data security.** Another challenge centers around fragmented data security issues, though new tools and technologies are surfacing. The Kerberos authentication protocol is a great step toward making Hadoop environments secure.
- **Full-fledged data management and governance.** Hadoop does not have easy-to-use, full-feature tools for data management, data cleansing, governance and metadata. Especially lacking are tools for data quality and standardization.

For big data and analytics, Hadoop is a life saver. Data gathered about people, processes, objects, tools, etc. is useful only when meaningful patterns emerge that, in-turn, result in better decisions. Hadoop helps overcome the challenge of the vastness of big data

5.1 Benefits of Hadoop for Big Data

Resilience — Data stored in any node is also replicated in other nodes of the cluster. This ensures fault tolerance. If one node goes down, there is always a backup of the data available in the cluster.

Scalability — Unlike traditional systems that have a limitation on data storage, Hadoop is scalable because it operates in a distributed environment. As the need arises, the setup can be easily expanded to include more servers that can store up to multiple petabytes of data.

Low cost — As Hadoop is an open-source framework, with no license to be procured, the costs are significantly lower compared to relational database systems. The use of inexpensive commodity hardware also works in its favor to keep the solution economical.

Speed — Hadoop's distributed file system, concurrent processing, and the MapReduce model enable running complex queries in a matter of seconds.

Data diversity — HDFS has the capability to store different data formats such as unstructured (e.g. videos), semi-structured (e.g. XML files), and structured.

While storing data, it is not required to validate against a predefined schema. Rather, the data can be dumped in any format. Later, when retrieved, data is parsed and fitted into any schema as needed. This gives the flexibility to derive different insights using the same data.

5.2 The Hadoop Ecosystem Supplementary Components

Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions.

There are four major elements of Hadoop i.e.

HDFS, MapReduce, YARN, and Hadoop Common.

Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.The following are a few supplementary components that are extensively used in the Hadoop ecosystem.

HDFS: Hadoop Distributed File System

YARN: Yet Another Resource Negotiator

MapReduce: Programming based Data Processing

Spark: In-Memory data processing

PIG, HIVE: Query based processing of data services

HBase: NOSQL Database

Mahout, Spark MLLib: MachineLearning algorithm libraries

Solar, Lucene: Searching and Indexing

Zookeeper: Managing cluster

Oozie: Job Scheduling

Apache Spark

It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.It consumes in memory resources hence, thus being faster than the prior in terms of optimization.Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably. Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably

PIG

Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.It is a platform for structuring the data flow, processing and analyzing huge data sets. Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the **JVM**. Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

HIVE

With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

Hbase

It's a NOSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google's BigTable, thus able to work on Big Data sets effectively.At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data.

Mahout

Mahout, allows Machine Learnability to a system or application. **Machine Learning**, as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

5.3 Other Components

Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

Solr, Lucene

These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.

Zookeeper

There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often. Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.

Oozie

Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs.i.e., Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

5.4 Open-Source software related to Hadoop

Open-Source Projects Related to Hadoop is our meritorious service with the aim of provides highly advanced and developed Hadoop projects for students and researchers in his globe. Today we are concentrating trendy of big data research concepts including energy efficient & green models, resource scheduling, sustainability issues, fault tolerance & reliability, machine learning techniques, graph analysis, large scale recommendation systems, index structures for big data analytics, exploratory analytics, big data management, scientific computing etc. Today, Hadoop is an Open-Source Tool that available in public. It is a framework that provides too many services like Pig, Impala, Hive, HBase, etc. The number of open-source tools growing in Hadoop ecosystem and these tools are continuously increasing. Let's view such open-source tools related to Hadoop,

Figure 1 Trendy of big data research concepts

Top Hadoop Related Open-Source Tools

1. Lucene [Java based text search engine]
2. Eclipse [Popular IDE written in Java]
3. HBase [Hadoop NOSQL Database]
4. Hive [Query data engine]
5. Jaql [Query language for JavaScript]
6. Pig [Large datasets analyzing platform]
7. Zookeeper [Centralized configuration service]
8. Avro [Data serialization system]
9. UIMA [unstructured analytic framework]
10. Presto [Distributed SQL query solution]

1. ***Lucene*** is an open-source Java based search library. It is very popular and a fast search library. It is used in Java based applications to add document search capability to any kind of application in a very simple and efficient way.Lucene is a simple yet powerful Java-based **Search** library. It can be used in any application to add search capability to it. Lucene is an open-source project. It is scalable. This high-performance library is used to index and search virtually any kind of text. Lucene library provides the core operations which are required by any search application. Indexing and searching.

How Search Application works?

Figure 2: How search application works?

Acquire Raw Content:

The initial stage in every search application is to gather the target contents on which the search will be performed.

Build the Document:

The next stage is to create the document from scratch, using material that the search programme can readily comprehend and analyse.

Analyse the Document:

Before beginning the indexing process, the document must be evaluated to determine whether parts of the text are candidates for indexing. This is the stage where the document is examined.

Indexing the Document:

Once documents are built and analyzed, the next step is to index them so that this document can be retrieved based on certain keys instead of the entire content of the document. The indexing method is comparable to indexes found at the end of books, where common terms are listed with page numbers so that they may be followed without having to search the whole book.

User Interface for Search:

After a database of indexes has been created, the programme may do any type of search. To make it easier for a user to conduct a search, the application must provide a means or user interface via which the user may enter text and initiate the search.

Build Query

When a user requests to search for a text, the application should create a query object based on that text, which may be used to query the index database for relevant information.

Search Query

The index database is then examined using a query object to obtain the necessary information and content documents.

Render Results

Once the result has been obtained, the programme must select how to provide the information to the user through the user interface. How much information should be displayed?

2. *Eclipse*

Eclipse is a Java IDE that is one of the 3 biggest and most popular IDE's in the world. It was written mostly in Java but it can also be used to develop applications in other programming languages apart from Java using plug-ins. Some of the features of Eclipse are as follows:

- PDE (Plugin Development Environment) is available in Eclipse for Java programmers that want to create specific functionalities in their applications.Eclipse flaunts powerful tools for the various processes in application development such as charting, modeling, reporting, testing, etc. so that Java developers can develop the application as fast as

possible.Eclipse can also be used to create various mathematical documents with LaTeX using the TeXlipse plug-in as well as packages for the Mathematica software.Eclipse can be used on platforms like Linux, macOS, Solaris and Windows.

3. HBase [Hadoop NOSQL Database]

HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data.

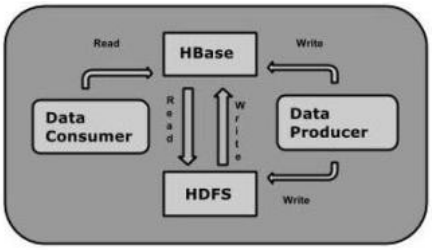


Figure 3 HBase [Hadoop NoSQL Database]

HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.

Storage Mechanism 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												

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:

4. Hive [Query Data Engine]

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.Hive is not

- A relational database. A design for OnLine Transaction Processing (OLTP). A language for real-time queries and row-level updates.
- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP. It provides SQL type language for querying called HiveQL or HQL. It is familiar, fast, scalable, and extensible.

Architecture of Hive

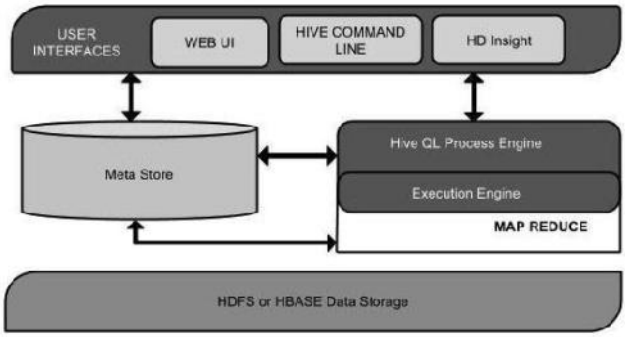


Figure 4 Architecture of Hive

User Interface

Hive is a data warehouse infrastructure programme that allows users to interface with HDFS. Hive provides three user interfaces: hive Web UI, Hive command line, and Hive HD insight (In windows server)

MetaStore

Hive selects appropriate database servers to hold the schema or metadata of tables, databases, table columns, data types, and HDFS mapping.

HiveQL Process Engine

HiveQL is a query language for the meta-store structure that is comparable to SQL. It is one of the MapReduce program's substitutes for the old technique. We can construct a query for a MapReduce task and process it instead of building a Java MapReduce application.

Execution Engine

The hive execution engine connects the HiveQL process engine to MapReduce. The query is processed by the execution engine, which generates the same MapReduce results. It makes use of the MapReduce flavor.

HDFS or HBASE

The data storage strategies used to store data in a file system are Hadoop distributed file system or HBASE.

5. *Jaql [Query language for JavaScript]*

JAQL is a query language for the JavaScript Object Notation (JSON) data interchange format. As its name implies, a primary use of JAQL is to handle data stored as JSON documents, but JAQL can work on various types of data. For example, it can support XML, comma-separated values (CSV) data and flat files. A "SQL within JAQL" capability lets programmers work with structured SQL data while employing a JSON data model that's less restrictive than its Structured Query Language counterparts.

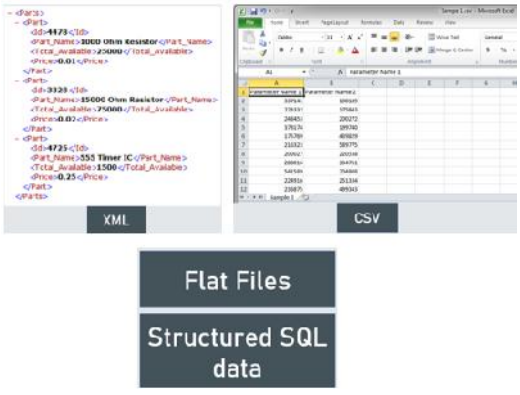


Figure 5Jaql [Query language for java script]

JSON has found wide use in Web and mobile applications, including large-scale big data and enterprise data warehouse applications. JAQL can run in local mode on individual systems and in cluster mode, in the latter case supporting Hadoop applications. It automatically generates MapReduce jobs and parallel queries on Hadoop systems. JAQL was created by workers at IBM Research Labs in 2008 and released to open source. While it continues to be hosted as a project on Google Code, where a downloadable version is available under an Apache 2.0 license, the major development activity around JAQL has remained centered at IBM. The company offers the query language as part of the tools suite associated with InfosphereBig Insights, its Hadoop platform. Working together with a workflow orchestrator, JAQL is used in Big Insights to exchange data between storage, processing and analytics jobs. It also provides links to external data and services, including relational databases and machine learning data.

6. *Pig [Large Datasets Analyzing Platform]*

Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with **Hadoop**; we can perform all the data manipulation operations in Hadoop using Apache Pig.To write data analysis programs, Pig provides a high-level language known as **Pig Latin**.

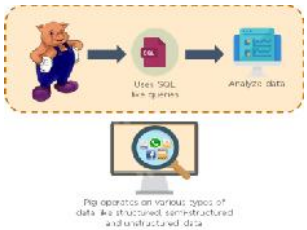


Figure 6Pig[large datasets analyzing platform]

This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data.To analyze data using **Apache Pig**, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a component known as **Pig Engine** that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.

Why Do We Need Apache Pig?

Programmers who are not so good at Java normally used to struggle working with Hadoop, especially while performing any MapReduce tasks. Apache Pig is a boon for all such programmers.

- **Pig Latin**

  Using **Pig Latin**, programmers can perform MapReduce tasks easily without having to type complex codes in Java.

- **Multi-query Approach**

  Apache Pig uses **multi-query approach**, thereby reducing the length of codes. For example, an operation that would require you to type 200 lines of code (LoC) in Java can be easily done by typing as less as just 10 LoC in Apache Pig. Ultimately Apache Pig reduces the development time by almost 16 times.

- **SQL-like Language**

  Pig Latin is **SQL-like language** and it is easy to learn Apache Pig when you are familiar with SQL.

- **Built-in Operators**

Apache Pig provides many built-in operators to support data operations like joins, filters, ordering, etc. In addition, it also provides nested data types like tuples, bags, and maps that are missing from MapReduce.

Features of Pig

Rich set of operators

− It provides many operators to perform operations like join, sort, filer, etc.

Ease of Programming

Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL.

Optimization Opportunities

The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language.

Extensibility

Using the existing operators, users can develop their own functions to read, process, and write data.

User-defined Functions

Pig provides the facility to create **User-defined Functions** in other programming languages such as Java and invoke or embed them in Pig Scripts.

Handles all Kinds of Data

Apache Pig analyzes all kinds of data, both structured as well as unstructured. It stores the results in HDFS.

7. *ZooKeeper*

Zookeeper is the easiest way for effective configuration management. It has two main benefits. First, it can be accessed from anywhere as it is stored centrally. This also reduces the issue with data integrity. Second, dynamic configuration management can be done as configuration data is stored centrally. This allows adjusting the system settings without restarting the system. Thus creating "znode" and storing configuration data is a handy way for configuration management.

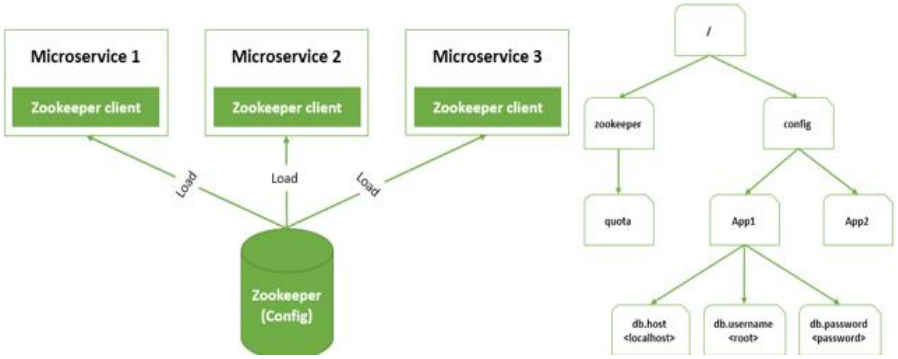


Figure 7 Zookeeper [Centralized configuration service]

This is a simplified version of how we are going to setup Zookeeper.Zookeeper stores data in a tree of ZNodes similar to Linux file system structure, a ZNode may contain another ZNodes or may have a value.**App1** and **App2** are sharing data from **/** and **config** znodes.However **db.host**, **db. username** and **db.password** are specific to **App1**.Zookeeper is one of the best centralized services for maintaining configuration, it is widely used by many other solutions like: Apache Hadoop, Kafka, SolrCloud

8. Avro [Data Serialization System]

Doug Cutting

Apache Avro is a language-neutral data serialization system. It was developed by Doug Cutting, the father of Hadoop. Since Hadoop writable classes lack language portability, Avro becomes quite helpful, as it deals with data formats that can be processed by multiple languages. Avro is a preferred tool to serialize data in Hadoop.Avro has a schema-based system. A language-independent schema is associated with its read and write operations. Avro serializes the data which has a built-in schema. Avro serializes the data into a compact binary format, which can be deserialized by any application.Avro uses JSON format to declare the data structures. Presently, it supports languages such as Java, C, C++, C#, Python, and Ruby.

Features of AVRO

- **language-neutral**
- processed by many languages
- **compressible** and **splittable**.
- **rich data structures**
- Avro **schemas** defined in **JSON**
- self-describing file named *Avro Data File*
- Remote Procedure Calls (RPCs).

Avro is a **language-neutral** data serialization system.It can be processed by many languages (currently C, C++, C#, Java, Python, and Ruby).Avro creates binary structured format that is both **compressible** and **splittable**. Hence it can be efficiently used as the input to Hadoop MapReduce jobs.Avro provides **rich data structures**. For example, you can create a record that contains an array, an enumerated type, and a sub record. These datatypes can be created in any language, can be processed in Hadoop, and the results can be fed to a third language.Avro **schemas** defined in **JSON**, facilitate implementation in the languages that already have JSON libraries.Avro creates a self-describing file named *Avro Data File,* in which it stores data along with its schema in the metadata section.Avro is also used in Remote Procedure Calls (RPCs). During RPC, client and server exchange schemas in the connection handshake.

9. UIMA [Unstructured Analytic Framework]

Unstructured Information Management applications are software systems that analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user.

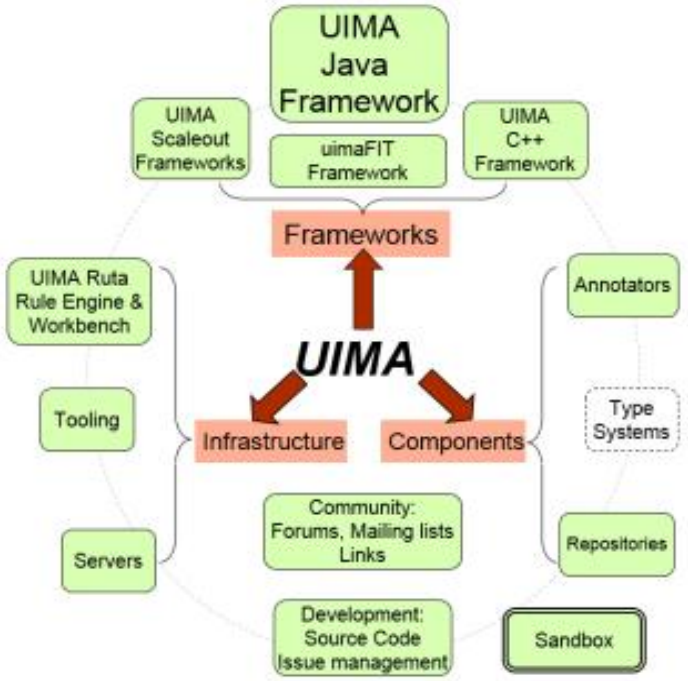


Figure 8UIMA[unstructured analytic framework]

An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at.UIMA enables applications to be decomposed into components, for example "language identification" => "language specific segmentation" => "sentence boundary detection" => "entity detection (person/place names etc.)". Each component implements interfaces defined by the framework and provides self-describing metadata via XML descriptor files. The framework manages these components and the data flow between them. Components are written in Java or C++; the data that flows between components is designed for efficient mapping between these languages.

10. *Presto [Distributed SQL Query Solution]*

Presto (or PrestoDB) is an open source, distributed SQL query engine, designed from the ground up for fast analytic queries against data of any size. It supports both non-relational sources, such as the Hadoop Distributed File System (HDFS), Amazon S3, Cassandra, MongoDB, and HBase, and relational data sources such as MySQL, PostgreSQL, Amazon Redshift, Microsoft SQL Server, and Teradata. Presto can query data where it is stored, without needing to move data into a separate analytics system. Query execution runs in parallel over a pure memory-based architecture, with most results returning in seconds. You'll find it used by many well-known companies like Facebook, Airbnb, Netflix, Atlassian, and Nasdaq.Presto is an open source, distributed SQL query engine designed for fast, interactive queries on data in HDFS, and others. Unlike Hadoop/HDFS, it does not have its own storage system. Thus, Presto is complimentary to Hadoop, with organizations adopting both to solve a broader business challenge. Presto can be installed with any implementation of Hadoop, and is packaged in the Amazon EMR Hadoop distribution.

5.5 Introduction to Big Data

Why Cloud Computing is the Answer to Your Big Data Initiatives. The cloud can help you process and analyze your big data faster, leading to insights that can improve your products and business.

Why Cloud Computing is the Answer to Your Big Data Initiatives

What are the Benefits to Companies from Advancement of Technology?

The advancement of technology has allowed companies to reap the benefits of streamlined processes and cost-efficient operations. But the one thing that has become a game changer for businesses of all sizes is the availability of data from every source imaginable – social media, sensors, business applications, and many more.These large stores of data that bombard companies day in and day out is collectively known as **big data**. Most have heard of it, many aim to maximize its potential to propel their business forward, and yet, only few have truly succeeded in doing so.At the same time, enterprises have adopted cloud computing to improve their IT operations and develop better software, faster.Merging big data with cloud computing is a powerful combination that can transform your organization.

What is Big Data?

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

The Concept of Big Data and What it Encompasses can be Better Understood with Four Vs:

- Volume

  The amount of data accumulated by private companies, public agencies, and other organizations on a daily basis is extremely large. This makes volume the defining characteristic for big data.

- Velocity

  It's a given that data can and will pile up really fast. But what matters is the speed with which you can process and examine this data so that it becomes useful information.

- Variety

  The types of data that get collected can be very diverse. Structured data contained in databases, and unstructured data such as tweets, emails, images, videos, and more, need to be consumed and processed all the same.

- Veracity

  Because of its scale and diversity, big data can contain a lot of noise. Veracity thus refers to the the certainty of the data and how your big data tools and analysis strategies can separate the poor quality data from those that really matter to your business.

- Technology leaders also name a fifth V – value. But this one isn't inherent within the huge amounts of raw data. Instead, the true value of big data can only be realized when the right information is captured and analyzed to gain actionable insights.To get a better idea of how big big data is, let's review some statistics:
- Over 1 billion Google searches are made and 294 billion emails are sent everyday
- Every minute, 65,972 Instagram photos are posted, 448,800 tweets are composed, and 500 hours-worth of YouTube videos are uploaded.
- By 2020, the number of smartphone users could reach 6.1 billion. And taking Internet of Things (IoT) into account, there could be 26 billion connected devices by then. For sure, big data is really big.

Why Should Big Data and its Exponential Growth Matter to your Business?

For one, an Accenture study (PDF) reveals that 79 percent of corporate executives surveyed believe that 'companies that do not embrace big data will lose their competitive position and may even face extinction'. Furthermore, an overwhelming 83 percent have taken on big data projects with the aim of outperforming others in their respective industries.Big data projects can impact almost any aspect of an organization. But as this survey by New Vantage Partners (PDF) shows, where it delivers most value to enterprises is in reducing costs (49.2%) and driving innovation (44.3%).

Big Data and the Cloud – a Match Made in Heaven?

Big data projects typically get started with data storage and application of basic analytics modules. However, as you discover ways to extract data at a much larger scale, you will need to find better methods to process and analyze this data, which will likely require infrastructure upgrades.You may add more capacity to your in-house data warehouse or power up more servers to cater to the rapidly-increasing analytics requirements. But even with the boost of your on-premise systems, your infrastructure eventually may not be able to keep up. This is where the cloud comes in, or more fittingly, when your big data goes to the cloud.This is where the cloud comes in, or more fittingly, when your big data goes to the cloud.

Why Big Data in the Cloud Makes Perfect Sense

The benefits of moving to the cloud are well documented. But these benefits take on a bigger role when we talk of big data analytics. Big data involves manipulating petabytes (and perhaps soon, exabytes and zettabytes) of data, and the cloud's scalable environment makes it possible to deploy data-intensive applications that power business analytics. The cloud also simplifies connectivity and collaboration within an organization, which gives more employees access to relevant analytics and streamlines data sharing.While it's easy for IT leaders to recognize the advantages of putting big data in the cloud, it may not be as simple to get C-suite executives and other primary stakeholders on board. But there's a business case to be made for the big data + cloud pairing because it gives executives a better view of the business and boosts data-driven decision making.For instance, optimization of the supply chain and efficient tracking of defects – both principal concerns of a COO of a physical product company – is made easier with material data on hand. Data is also key for the CMO looking to increase customer engagement and loyalty, and for the CFO seeking new opportunities for cost reduction, revenue growth, and strategic investments.And all of these insights can be easily presented to the CEO to inform fast, strategic decision making.Whatever perspective you may have, big data complemented with an agile cloud platform can affect significant change in the way your organization does business and achieves your objectives.

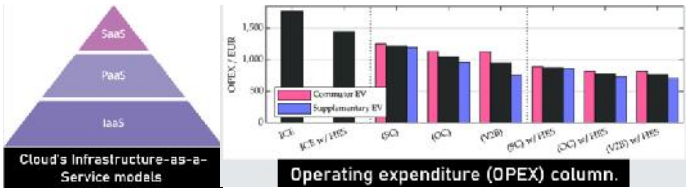
Many enterprises are already making the move. A Forrester Research survey in 2017 revealed that big data solutions via cloud subscriptions will increase about 7.5 times faster than on-premise options.

5.6 Big Opportunities, Big Challenges

Bringing big data to the cloud presents huge opportunities, but there are some challenges that need to be overcome. Let's go over the advantages first.

Requires zero CAPEX

The cloud has fundamentally changed IT spending as organizations know it—and in a good way.As we mentioned earlier, big data projects require immense infrastructure resources, which traditionally would also mean high on-premise capital expenditure (CAPEX) investments. But the cloud's Infrastructure-as-a-Service models have allowed companies to practically eliminate its biggest CAPEX expenses by shifting these into the operating expenditure (OPEX) column. So, when you need to set up your database servers or data warehouses, you won't need to make massive upfront investments. This has been one of the most compelling benefits that has convinced businesses to migrate to the cloud.



Enables Faster Scalability

Large volumes of both structured and unstructured data requires increased processing power, storage, and more. The cloud provides not only readily-available infrastructure, but also the ability to scale this infrastructure really quickly so you can manage large spikes in traffic or usage.

Lowers the Cost of Analytics

Mining big data in the cloud has made the analytics process less costly. In addition to the reduction of on-premise infrastructure, you can also save on costs related to system maintenance and upgrades, energy consumption, facility management, and more. You can also worry less about the technical aspects of processing big data and focus more on creating insights. Even better, the cloud's pay-as-you-go model is more cost-efficient, with little waste of resources.

Encourages an Agile and Innovative Culture

The ability to innovate is a mindset that should be cultivated within any enterprise. This type of culture can lead to creative ways of using big data to gain a competitive advantage, and the cloud makes it easier to spin up the necessary infrastructure to do so. When your team focuses on analyzing data instead of managing servers and databases, you can more easily and quickly unearth insights that can help you augment product lines, boost operational efficiency, improve customer service, and more.

Enables Better Business Continuity and Disaster Recovery

In cases of cyber-attacks, power outages or equipment failure, traditional data recovery strategies will no longer do the trick. The task of replicating a data center – with duplicate storage, servers, networking equipment, and other infrastructure – in preparation for a disaster is tedious, difficult, and expensive.In addition, legacy systems often take very long to back up and restore. This is especially true in the era of big data, when data stores are so immense and expansive.Having the data stored in cloud infrastructure will allow your organization to recover from disasters faster, thus ensuring continued access to information and vital big data insights.

5.7 Potential Challenges of Big Data in the Cloud

Migrating big data to the cloud presents various hurdles. Overcoming these require a concerted effort from IT leaders, C-suite executives, and other business stakeholders. Here are some of the major challenges of big data cloud implementations.These large datasets often contain sensitive information such as individuals' addresses, credit card details, social security numbers, and other personal information.

Less Control Over Security

These large datasets often contain sensitive information such as individuals' addresses, credit card details, social security numbers, and other personal information. Ensuring that this data is kept protected is of paramount importance. Data breaches could mean serious penalties under various regulations and a tarnished company brand, which can lead to loss customers and revenue.While security should not be a hindrance to migrating to the cloud, you will have less direct control over your data, which can be a big organizational change and may cause some discomfort.To deal with this, be sure to carefully evaluate the security protocols and understand the shared responsibility model of your cloud service provider so you know what your roles and obligations are.

Less Control Over Compliance

Compliance is another concern that you'll have to think about when moving data to the cloud.Cloud service providers maintain a certain level of compliance with various regulations such as HIPAA, PCI, and many more. But similar to security, you no longer have full control over your data's compliance requirements.

Even if your CSP is managing a good chunk of your compliance, you should make sure you know the answers to the following questions:

- Where is the data going to reside?
- Who is going to manage it, and who can access it?
- What local data regulations do I need to comply with?
- If your company is in a highly regulated industry like healthcare or finance, these questions become much more important.
- Make sure you know exactly what data is stored where, ensure that your CSP has robust compliance policies, understand the shared responsibility model, potentially create Service Level Agreements (SLAs) for compliance.

Network Dependency and Latency Issues

The flipside of having easy connectivity to data in the cloud is that availability of the data is highly reliant on network connection. This dependence on the internet means that the system could be prone to service interruptions.In addition, the issue of latency in the cloud environment could well come into play given the volume of data that's being transferred, analyzed, and processed at any given time.Big data doesn't have to equal big chaos. Yes, the volume and the speed with which data is growing can be overwhelming, especially for organizations just starting out. But by utilizing the cloud for big data initiatives, your enterprise can transform itself into an efficient, data-driven organization.

5.8 Overview of Big Data

Due to the advent of new technologies, devices, and communication means like social networking sites, the amount of data produced by mankind is growing rapidly every year.The amount of data produced by us from the beginning of time till 2003 was 5 billion gigabytes. If you pile up the data in the form of disks it may fill an entire football field. The same amount was created in every two days in 2011, and in every ten minutes in 2013. This rate is still growing enormously. Though all this information produced is meaningful and can be useful when processed, it is being neglected.

What is Big Data?

Big data is a collection of large datasets that cannot be processed using traditional computing techniques.It is not a single technique or a tool, rather it has become a complete subject, which involves various tools, techniques and frameworks.

What Comes Under Big Data?

- **Black Box Data**- It is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.
- **Social Media Data**-The social media such as Facebook and Twitter hold information and the views posted by millions of people across the globe.
- **Stock Exchange Data** − The stock exchange data holds information about the 'buy' and 'sell' decisions made on a share of different companies made by the customers.
- **Power Grid Data** − The power grid data holds information consumed by a particular node with respect to a base station.
- **Transport Data** − Transport data includes model, capacity, distance and availability of a vehicle.
- **Search Engine Data** − Search engines retrieve lots of data from different databases.

Thus, Big Data includes huge volume, high velocity, and extensible variety of data

- **Structured Data** − Relational data.
- **Semi Structured Data** − XML data.
- **Unstructured Data** − Word, PDF, Text, Media Logs.

5.9 Benefits of Big Data

Using the information kept in the social network like Facebook, the marketing agencies are learning about the response for their campaigns, promotions, and other advertising mediums.Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.Using the data regarding the previous medical history of patients, hospitals are providing better and quick service.

Big Data Technologies

To harness the power of big data, you would require an infrastructure that can manage and process huge volumes of structured and unstructured data in realtime and can protect data privacy and security.There are various technologies in the market from different vendors including Amazon, IBM, Microsoft, etc., to handle big data. While looking into the technologies that handle big data, we examine the following two classes of technology –

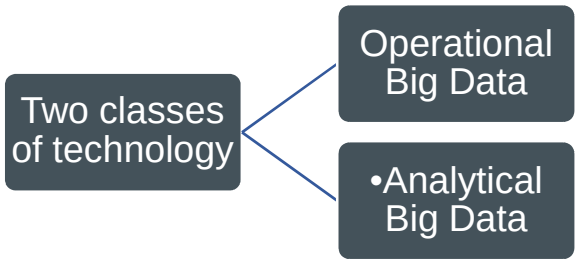


Figure 9 Big Data Technologies

Operational Big Data

This includes systems like MongoDB that provide operational capabilities for real-time, interactive workloads where data is primarily captured and stored. NOSQL Big Data systems are designed to take advantage of new cloud computing architectures that have emerged over the past decade to allow massive computations to be run inexpensively and efficiently. This makes operational big data workloads much easier to manage, cheaper, and faster to implement. Some NOSQL systems can provide insights into patterns and trends based on real-time data with minimal coding and without the need for data scientists and additional infrastructure.

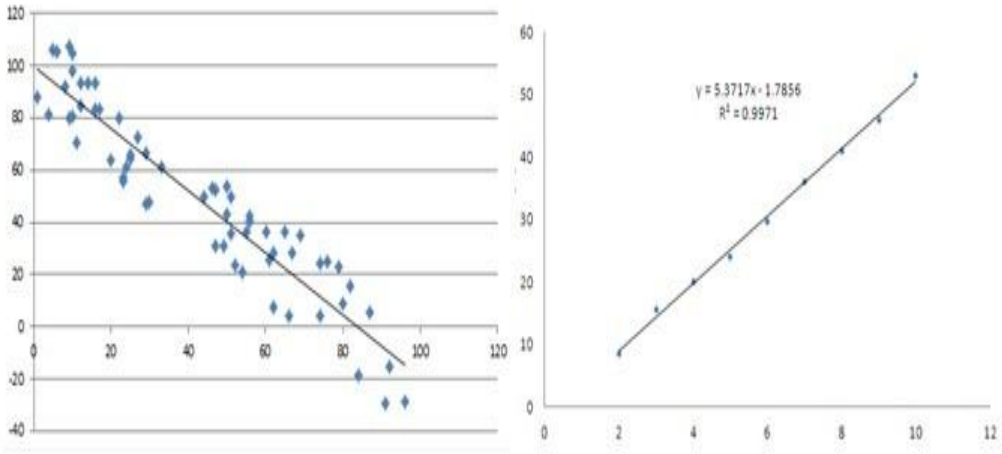


Figure 10 Patterns and Trends

Analytical Big Data

These includes systems like Massively Parallel Processing (MPP) database systems and MapReduce that provide analytical capabilities for retrospective and complex analysis that may touch most or all of the data.MapReduce provides a new method of analyzing data that is complementary to the capabilities provided by SQL, and a system based on MapReduce that can be scaled up from single servers to thousands of high- and low-end machines.

These two classes of technology are complementary and frequently deployed together.

Operational vs Technical System

	Operational	Analytical
Latency	1 ms - 100 ms	1 min - 100 min
Concurrency	1000 - 100,000	1 - 10
Access	Writes and	Reads

Pattern	Reads	
Queries	Selective	Unselective
Data Scope	Operational	Retrospective
End User	Customer	Data Scientist
Technology	NoSQL	MapReduce, MPP Database

5.10 Big Data Challenges

Many businesses become stuck in the early stages of their Big Data efforts. This is due to the fact that they are neither aware of nor able to deal with the problems posed by Big Data.

- **Lack of** Proper **Understanding of Big Data**

Insufficient knowledge causes companies to fail in their Big Data endeavors. Employees may not understand what data is, how it is stored, processed, and where it comes from. Others may not have a clear picture of what is going on, even if data specialists do.

Employees who do not understand the value of data storage, for example, may fail to preserve a backup of important data. They may not be appropriately storing data in databases. As a result, when this critical information is needed, it is difficult to locate.

Solution

Workshops and seminars on big data should be offered at firms for everyone. All staff that handle data on a regular basis and are involved in Big Data projects should receive basic training. All levels of the company must have a fundamental awareness of data ideas.

- **Data Growth Issues**

One of the most serious Big Data concerns is appropriately storing all of these massive volumes of data. The amount of data kept in data centres and company databases is continually expanding. It becomes increasingly challenging to manage big data sets as they increase rapidly over time.

The majority of the data is unstructured and originates from a variety of sources, including documents, movies, audios, text files, and other media. This implies they aren't searchable in databases.

Solution

Companies are using current approaches like compression, tiering, and deduplication to handle these massive data collections. Compression reduces the number of bits in data, resulting in a smaller total size. The process of deleting duplicate and unnecessary data from a data set is known as deduplication.

Companies can store data in separate storage levels via data tiering. It guarantees that the data is stored in the best possible location. Depending on the size and relevance of the data, data tiers might include public cloud, private cloud, and flash storage.

Companies are also turning to Big Data technologies like Hadoop, NOSQL, and others.

This brings us to the third issue with Big Data.

- **Confusion While Big Data Tool Selection**

Companies frequently become perplexed while deciding on the right instrument for Big Data analysis and storage. Is HBase or Cassandra the superior data storage technology? Will Spark be a better solution for data analytics and storage than Hadoop MapReduce?

Companies are bothered by these problems, and they are sometimes unable to find solutions. They end up making poor choices and choosing the wrong technologies. As a result, resources such as money, time, effort, and work hours are squandered.

Solution

The best course of action is to seek expert assistance. You may either engage seasoned specialists who are far more knowledgeable about these instruments. Another option is to hire Big Data consultants. Consultants will provide recommendations for the appropriate tools based on the situation at your firm. You may devise a plan and then choose the ideal instrument for you based on their recommendations.

- **Lack of Data Professionals**

Companies require trained data specialists to run these contemporary technology and Big Data solutions. Data scientists, data analysts, and data engineers who have worked with the tools and made sense of large data volumes will be among these experts.

Companies are facing a scarcity of Big Data experts. This is due to the fact that data processing tools have advanced fast, but most experts have not. In order to close the gap, concrete efforts must be done.

Solution

Companies are devoting greater resources to the recruitment of talented workers. They must also provide training programmes for current employees in order to get the most out of them.

Another key move made by businesses is the procurement of artificial intelligence/machine learning-powered data analytics solutions. These tools may be used by professionals who aren't data scientists but have a rudimentary understanding of the subject. This stage allows businesses to save a significant amount of money on recruitment.

Securing Data

One of the most difficult aspects of Big Data is securing these massive data collections. Companies are frequently so preoccupied with comprehending, preserving, and analyzing their data sets that data security is pushed to the back burner. Unprotected data stores, on the other hand, may become breeding grounds for malevolent hackers.

A stolen record or a data breach may cost a company up to $3.7 million.

Solution

To secure their data, businesses are hiring more cybersecurity workers. Other measures made to protect data include:

Encrypting data

Separation of data

Control of identity and access

Endpoint security implementation

Security monitoring in real time

Make use of Big Data security technologies like IBM Guardian.

Summary

- Apache Hadoop is a set of open-source software tools for solving issues involving large volumes of data and processing utilising a network of many computers. It's a MapReduce programming model-based software framework for distributed storage and processing of massive data.
- Big data refers to massive, difficult-to-manage data quantities – both organised and unstructured – that inundate enterprises on a daily basis. Big data may be evaluated for insights that help people make better judgments and feel more confident about making key business decisions.

- HDFS, or Hadoop Distributed File System, is a distributed file system that runs on commodity hardware. It has a lot in common with other distributed file systems. However, there are considerable distinctions between it and other distributed file systems. HDFS is meant to run on low-cost hardware and is extremely fault-tolerant. HDFS is a file system that allows high-throughput access to application data and is well-suited to applications with huge data collections. To provide streaming access to file system data, HDFS relaxes a few POSIX criteria.
- In a Hadoop cluster, MapReduce is a programming paradigm that permits tremendous scalability over hundreds or thousands of computers. MapReduce, as the processing component, lies at the heart of Apache Hadoop
- Hadoop Ecosystem is a platform or a suite that offers a variety of services to address big data issues. It consists of Apache projects as well as a variety of commercial tools and solutions. HDFS, MapReduce, YARN, and Hadoop Common are the four core components of Hadoop.
- Apache Pig is a high-level framework for developing Hadoop-based apps. Pig Latin is the name of the platform's language. Pig's Hadoop tasks may be run in MapReduce, Apache Tez, or Apache Spark.
- Eclipse is a robust Java programming environment. Because Hadoop and Mapreduce programming is done in Java, we should use an Integrated Development Environment with a lot of features (IDE)
- Jaql is one of the languages used to abstract the intricacies of Hadoop's MapReduce programming architecture. It's a functional language with a weakly typed syntax and lazy evaluation.

Keywords

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

BigData: Big Data is a massive collection of data that continues to increase dramatically over time. It is a data set that is so huge and complicated that no typical data management technologies can effectively store or process it. Big data is similar to regular data, except it is much larger.

HDFS: Hadoop File System was built on distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Name Node: The name node is a piece of commodity hardware that houses the GNU/Linux operating system as well as name node software. It's a piece of software that can run on standard hardware.

Data Node: The data node is a commodity computer with the GNU/Linux operating system and data node software installed. In a cluster, there will be a data node for each node (common hardware/system).

HBase: HBase is a Hadoop-based open-source database with sorted map data. It's horizontally scalable and column-oriented.

JAQL: Any software package that is used in connection with databases for searching, processing, or even generating JavaScript Object Notion (JSON)-based documents is known as JSON query language (JAQL).

Self Assessment

Q1: A parallel computer system can do a lot of things.

A. Decentralized computing
B. Parallel computing
C. Centralized computing
D. All of these

Q2: The process of creating parallel programs is known _____

A. Parallel computation
B. Parallel processes
C. Parallel programming
D. Parallel development

Q3: Pig is mostly concerned with ___ number of nodes.

A. Two
B. Three
C. Four
D. Five

Q4: Pig was basically developed by _____

A. Twitter
B. Facebook
C. Google
D. Yahoo

Q5: HIVE performs _____ and _____ of large data sets.

A. Reading and writing
B. Execution and writing
C. Both
D. None of above

Q6:Hadoop is an _____ that available in public

A. Open-Source tool
B. Commercial tool
C. House tool
D. Vendor tool

Q7: Hadoop is a framework that provides too many services like

A. Pig
B. HBase
C. Hive
D. All of above

Q8: Top Hadoop Related Open-Source Tools are

A. Hive
B. Jaql
C. Pig
D. All of above

Q9: Lucene is a simple yet powerful Java-based _____ library

A. Search
B. Reporting
C. Both
D. None of above

Q10: _____ is a Java IDE that is one of the 3 biggest and most popular IDE's in the world

A. Paint
B. Notebook
C. Eclipse
D. All of above

Q11: The concept of big data and what it encompasses can be better understood with four Vs. Those

are:

A. Volume
B. Velocity
C. Veracity
D. All of above

Q12: _____ refers to the certainty of the data and how your big data tools and analysis strategies can separate the poor-quality data from those that really matter to your business.

A. Volume
B. Velocity
C. Veracity
D. All of above

Q13: Data in Big Data is _____ bytes in size.

A. Terra
B. Mega
C. Giga
D. Peta

Q14: Select the types of Big Data

A. Structured Data
B. Unstructured Data
C. Semi-structured Data
D. All of the above

Q15: _____ is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.

A. Social media data
B. Black box data
C. Stock exchange data
D. Power grid data

Answers for Self Assessment

1.	A	2.	C	3.	A	4.	D	5.	A
6.	B	7.	D	8	D	9	A	10	C
11	D	12	A	13	D	14	D	15	B

Review Questions

1. Difference between data mart and data ware house.
2. Writedown the Tips for Creating Effective Big Data Models.
3. Explain different types of data mart.
4. Write down advantages and disadvantages of data mart.
5. What do you understand by data streaming? Explain Use Cases for Real-Time and Streaming Data.

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
- of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NOSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 06: Hadoop Administration

CONTENTS

Objectives

- Learn Hadoop installation step by step
- Learn HDFS
- Learn about HDFS Architecture
- Learn Goals of HDFS
- Learn basic commands in HDFS

Introduction

Hadoop is primarily supported by the Linux operating system and its features. If you're using Windows, you can use Cloudera VMware, which comes with Hadoop preconfigured, or Oracle VirtualBox, or VMware Workstation. In this chapter, we will learn how to install Hadoop on VMware Workstation 12 using VMware Workstation. This will be accomplished by installing CentOS on my virtual machine.

Prerequisites

You can use any of these to install the operating system.

VirtualBox/VMWare/Cloudera

You can use any of these to install the operating system.

Operating System:

On Linux-based operating systems, Hadoop may be installed. Ubuntu and CentOS are two of the most popular operating systems among them. We'll be using CentOS for this course.

Java

On your computer, you must install the Java 8 package.

Hadoop

The Hadoop 2.7.3 package is required.

Hadoop Installation on Windows

Note: If you're using Linux, go straight to Step 9.

Step1: Installing VMware Workstation

- You can download the VMWare workstation by using the below link

https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_pro/15_0

- Open the.exe file after it has been downloaded and change the path to the desired location.
- Follow the installation instructions to the letter.

Step2: Installing CentOS

- Use the following link to download CentOS.
  https://www.centos.org/download/
- Save the file wherever you choose.

Step3: Using VMware to install CentOS

The following window appears when you launch VMware:

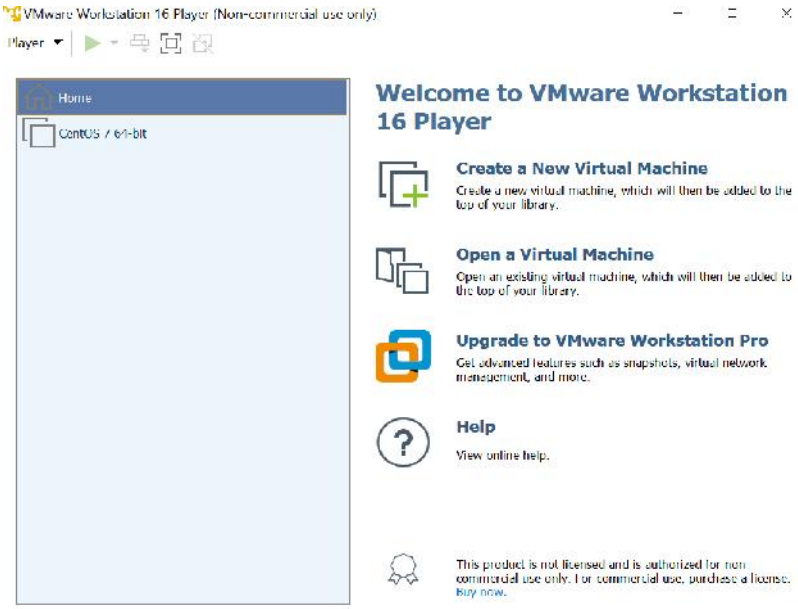


Figure 1: VMWare Workstation

Select Create a New Virtual Machine from the drop-down menu as shown in Figure 2.

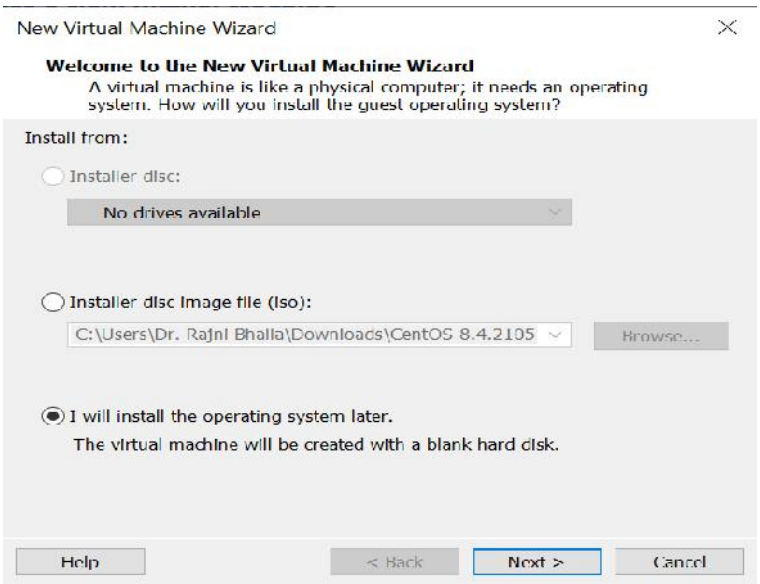


Figure 2: New Virtual Machine Wizard

1. Browse to the location of the CentOS file you downloaded, as seen in the image above. It is important to note that it must be a disc image file.
2. Click on Next
3. Choose the name of your machine.
4. Then, click Next

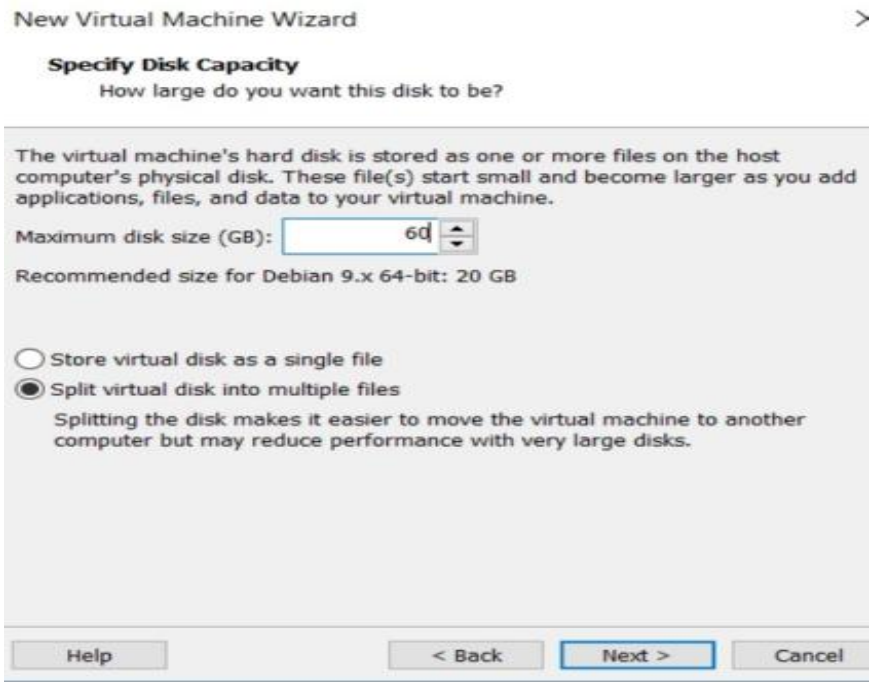


Figure 3: Specify disk capacity

5. Specify the disk capacity.
6. Click next
7. Click on Finish

Figure 4: Options

8. You may see three options in the image above: I Finished Installing, Change Disc, and Help. You don't have to touch any of them until your CentOS installation is complete.
9. Your system is currently being tested and prepared for installation as shown in Figure 5.

Figure 5: Installation in process

10. When the checking percentage hits 100%, you will be brought to the following screen:

Figure 6: Installation Summary

11. You may select your preferred language here. English is the default language, and that is what I have chosen. Then, click on continue.

Figure 7 Select Language

12. If you wish to pick a different language, select it
13. Click on Continue.

Step4:

The login screen will look like this:

Enter the user ID and password you created before.

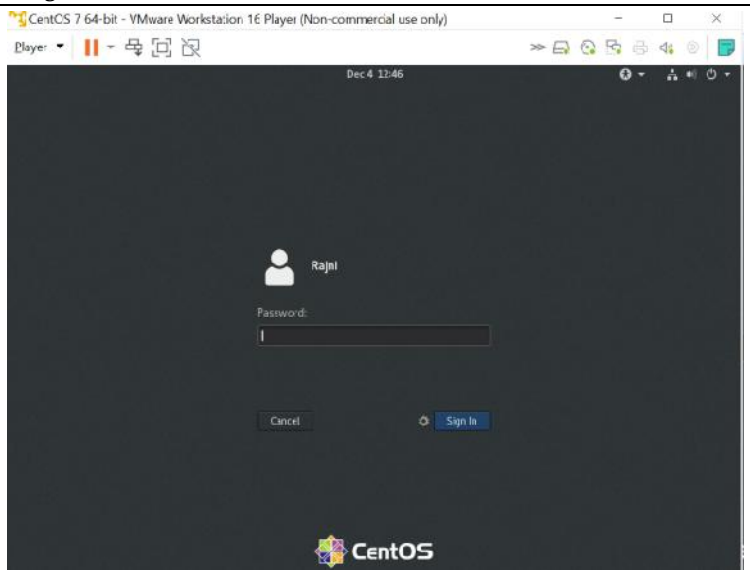


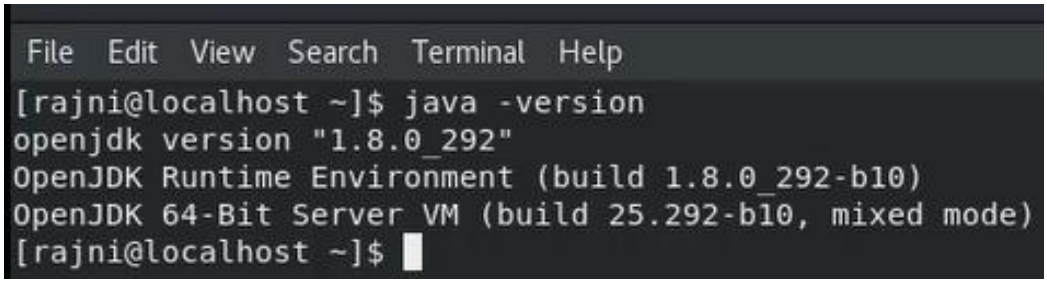
Figure 8: Login

The installation of CentOS is now complete!

You must now begin working on CentOS rather than your local operating system. If you've skipped forward to this step because you're already using Linux/Ubuntu, proceed to the next step.

All commands must be entered into the Terminal. By right-clicking on the desktop and selecting Open Terminal, you may access the Terminal.

Step 5: We will use the command java - version to check whether java is already installed or not.



Step 6: Downloading and Installing Java 8

- The Java 8 Package may be downloaded by clicking here. This file should be saved in your home directory.
- Using the following command, extract the Java tar file:

tar -xvf jdk-8u101-linux-i586.tar.gz

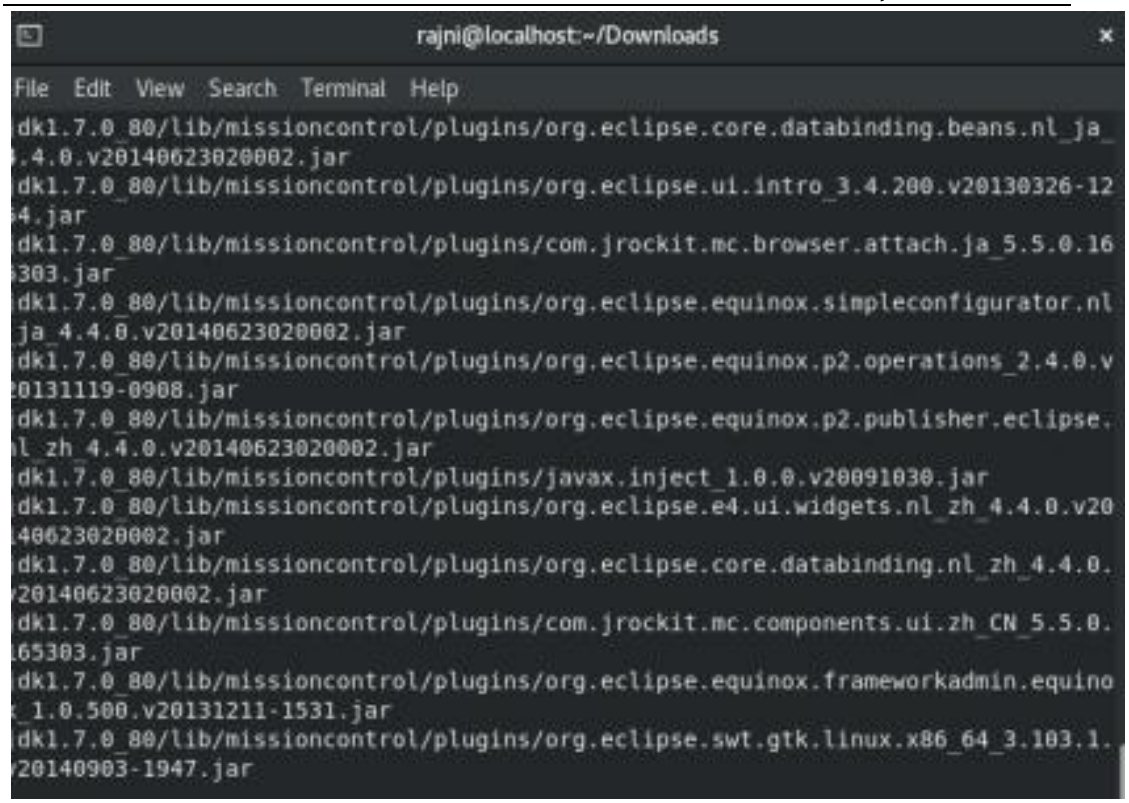


Figure 9: JDK installation in process

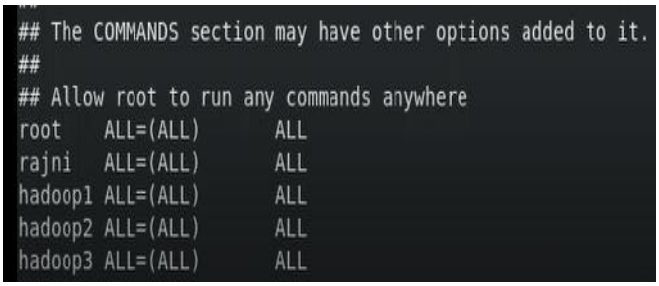
Step 7: Command to create a Sudo user.

We will create a Sudo user before installing Hadoop.

Step 8: To make an entry on sudoers file for Hadoop users. **edits the sudoers file**, which is used by the sudo command. To change what users and groups are allowed to run sudo, run visudo

$ visudo

We want hadoop3 must be allowed to run any command anywhere.



Step 9: Set up key-based ssh to its account.ssh-keygen is a tool for creating new authentication key pairs for SSH. Such key pairs are used for automating logins, single sign-on, and for authenticating hosts. This is the key you need to copy into your remote device to get successful SSH authentication. Generating public/private RSA key pair.Remember to set the correct permissions on the file system using command chmod.To authenticate to a remote server we should now copy the generated public key in a specific location on the remote server, which is the authorized_keys file under ~/.ssh folder.

```
[rajni@localhost ~]$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
/home/rajni/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/rajni/.ssh/id_rsa.
Your public key has been saved in /home/rajni/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:pJ9+E/Gju1/3oHQTpDvsKehiUamlFiFHP0I/6Tqewlc rajni@localhost.localdomain
The key's randomart image is:
+---[RSA 3072]----+
|      .o         |
|     ..oo .      |
|     o..B.    .  |
|      .=+o.  o   |
|      .*S  o. .  |
|      =oE...o. . |
|    . .ooo. o=.= .|
|     o.++. =o * +.|
|      +ooo.o*=   .|
+----[SHA256]-----+
[rajni@localhost ~]$ cp -r ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
[rajni@localhost ~]$ chmod 0600 ~/.ssh/authorized_keys
[rajni@localhost ~]$
```

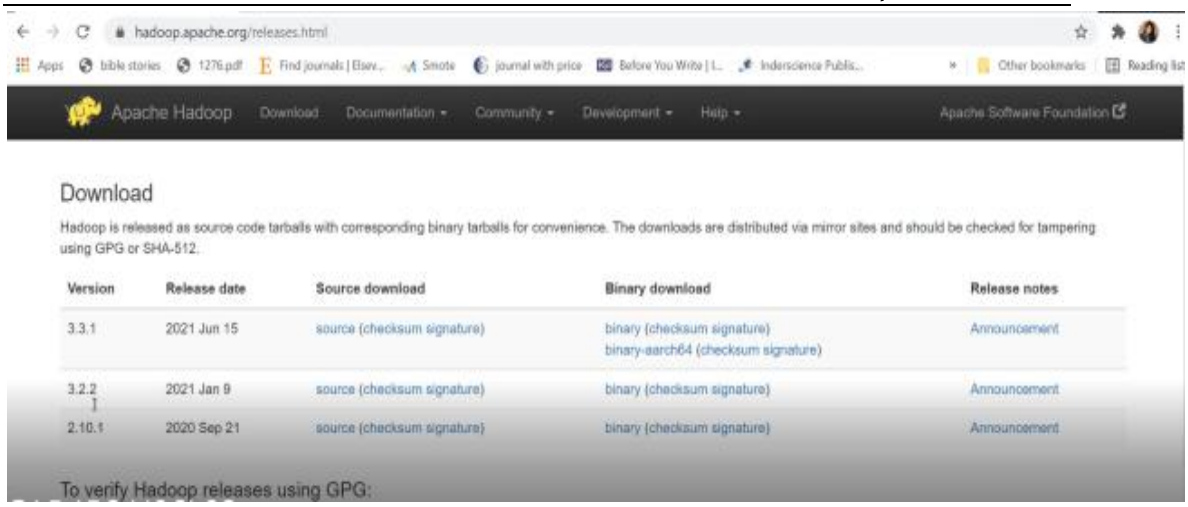
Step10: Verify key based login by using following command

```
File  Edit  View  Search  Terminal  Help
[rajni@localhost ~]$ ssh localhost
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Jul 23 15:38:16 2021
[rajni@localhost ~]$
```

Step11:Downloading and installing Hadoop

https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.10.1/hadoop-3.2.2.tar.gz

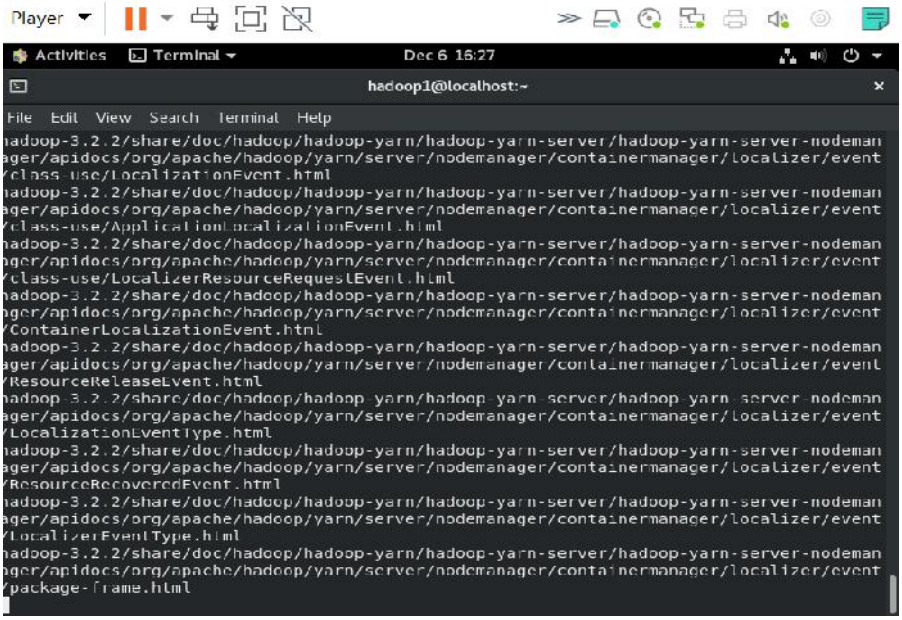


Make sure to download the binary file.

On the terminal, use the following command to extract the Hadoop file:

tar -xvf hadoop-3.2.2.tar.gz

Extracting hadoop file as shown in figure below:



Step12: Moving Hadoop to a Location

After using the ls command,we will notice Hadoop folder is created. In next step, we will rename the file using the mv command from Hadoop-3.2.2 to Hadoop.

Step13: Editing and Configuring Hadoop you must first set the path in the ~/.bashrc file. The command ~ /.bashrc can be used to set the path from the root user. You should check your Java configurations before editing ~/.bashrc.

update-alternatives-config java

You'll now be able to see all of the Java versions installed on the computer. Because I only have one version of Java, which is the most recent, it is displayed below:

You can also have several versions.

The next step is to choose the version you wish to work on. In the screenshot above, you can see a path that has been marked. Copy and paste this path into a gedit file. This route will only be utilised in the next phases.

Enter the number of the selection you've made. I've picked number one in this case.Now use the vi editor to open /.bashrc (the screen-oriented text editor in Linux)

Note that you must first become a root user before editing /.bashrc.

When you get logged into your root user, enter the command:

vi ~/.bashrc

The command above should bring you to the vi editor, where you should see the following screen:

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

To get to this, hit the Insert key on your keyboard, and then start typing the following code to set a Java path:

fi

#HADOOP VARIABLES START

export JAVA_HOME= (path you copied in the previous step)

export HADOOP_HOME=/home/(your username)/hadoop

export PATH=$PATH:$HADOOP_HOME/bin

export PATH=$PATH:$HADOOP_HOME/sbin

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

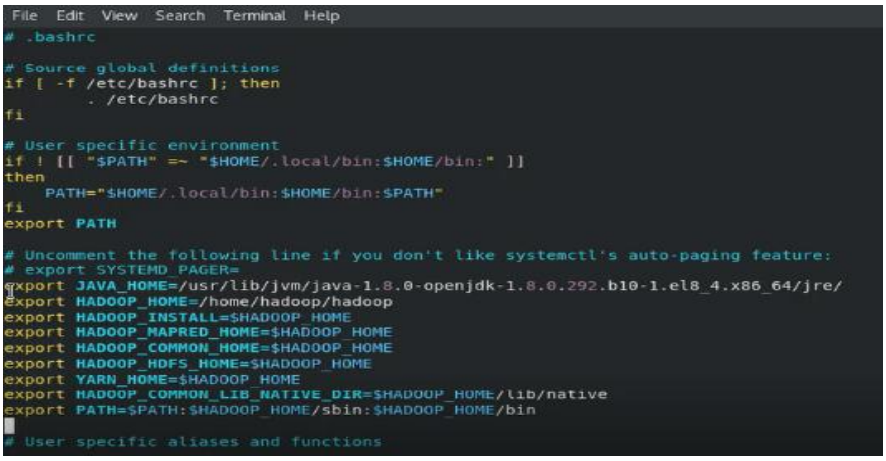
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_/HOME/lib/native

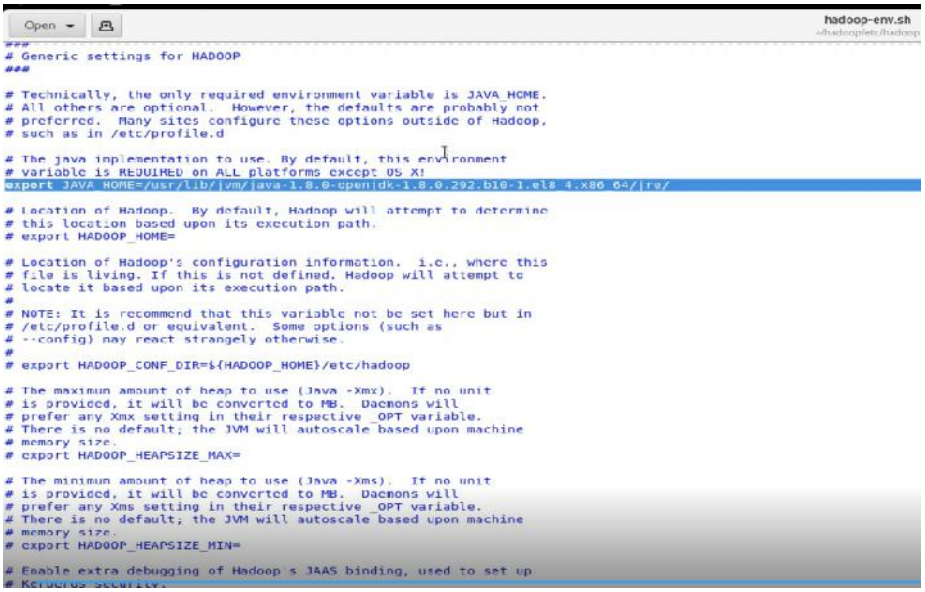
export HADOOP_OPTS="Djava.library.path"=$HADOOP_HOME/lib"

#HADOOP VARIABLES END

After writing the code, click on **Esc** on your keyboard and write the command: wq**!** This will save and exit you from the vi editor. The path has been set now as it can be seen in the image below:

Step14: Using the vi editor, open hadoop-env.sh. To inform Hadoop which path to use, replace this path with the Java path. You will be presented with the following window:





Step15: There are multiple XML files that need to be modified now, and you must specify the property and path for each one. All configuration files are shown in the image below:

- **Editing core-site.xml**
  - use the command below to open core-site.xml file

    vim core-site.xml
  - Enter the following code in between the configuration tags as below

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
        <name>hadoop.tmp.dir</name>
        <value>file:///var/hadoop_warehouse/tmp/hadoop</value>
</property>
<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000 </value>
</property>

</configuration>
```

  - Now, exit from this window by entering the command: wq!

- **Editing hdfs-site.xml**
  - Use the command below to open hdfs-site.xml

vim hdfs-site.xml

- Enter the following code in between the configuration tags as below

```
<configuration>
<property>
        <name>dfs.replication</name>
        <value>1</value>
</property>
<property>
        <name>dfs.namenode.name.dir</name>
        <value>file:///var/hadoop_warehouse/dfs/namenode</value>
</property>
<property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///var/hadoop_warehouse/dfs/datanode</value>
</property>
<property>
        <name>dfs.datanode.checkpoint.dir</name>
        <value>file:///var/hadoop_warehouse/dfs/secondary</value>
</property>

<configuration>
```

- Exit using **Esc** and the command: wq**!**

- **Editing mapred-site.xml**
  - Use the command below to open hdfs-site.xml

    vim mapred-site.xml
  - Enter the following code in between the configuration tags as below:

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
        <name>mapreduce..job.tracker</name>
        <value>localhost:9001</value>
</property>
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
</property>
<property>
 <name>mapreduce.application.classpath</name>
 <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
</property>
</configuration>
```

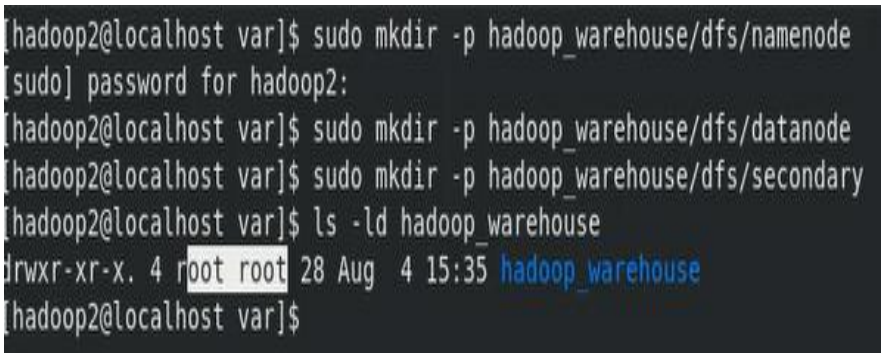
Exit using **Esc** and the command: wq**!**

- **Editing yarn-site.xml**
  - Use the command below to open hdfs-site.xml

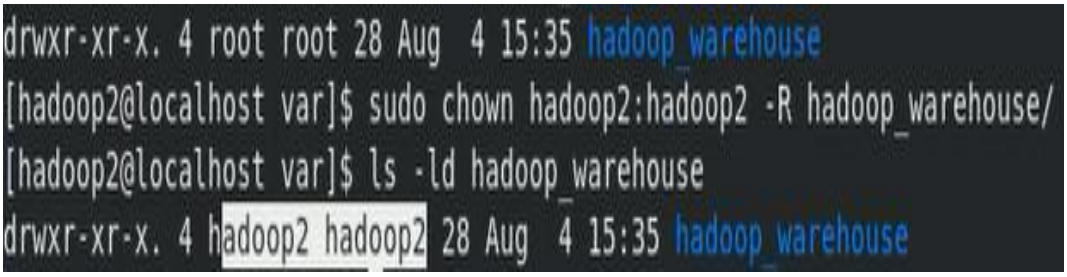
vim yarn-site.xml

- Exit from this window by pressing Esc and then writing the command: wq**!**

Step16: Create a directory namenode, datanode, and secondary using the command below:



Step 17: As we can see in the above image, permissions have been given only to the root. So, next, we will use chown command as shown in image below to change permission to hadoop2.



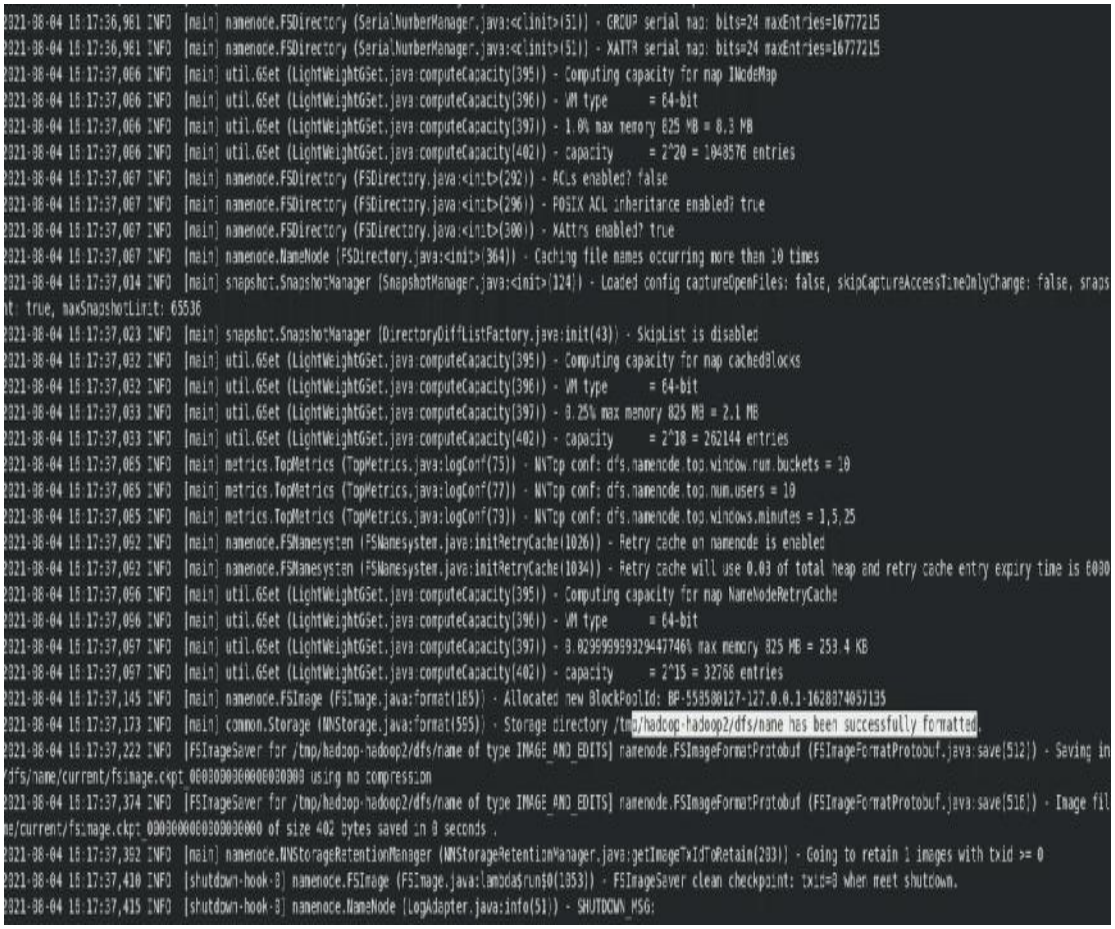
Step 18: To check permissions of all the file that comes under Hadoop_datawarehouse, following command will be executed:

```
[hadoop2@localhost var]$ cd hadoop_warehouse
[hadoop2@localhost hadoop_warehouse]$ ls
dfs  tmp
[hadoop2@localhost hadoop_warehouse]$ ll
total 0
drwxr-xr-x. 5 hadoop2 hadoop2 55 Aug  4 15:35 dfs
drwxr-xr-x. 3 hadoop2 hadoop2 20 Aug  4 15:26 tmp
[hadoop2@localhost hadoop_warehouse]$
```

All the files that comes into this folder their permission has changed.

Step22: Lets go to the Hadoop directory and run the command as shown below to format the name node.

hadoopnamenode -format

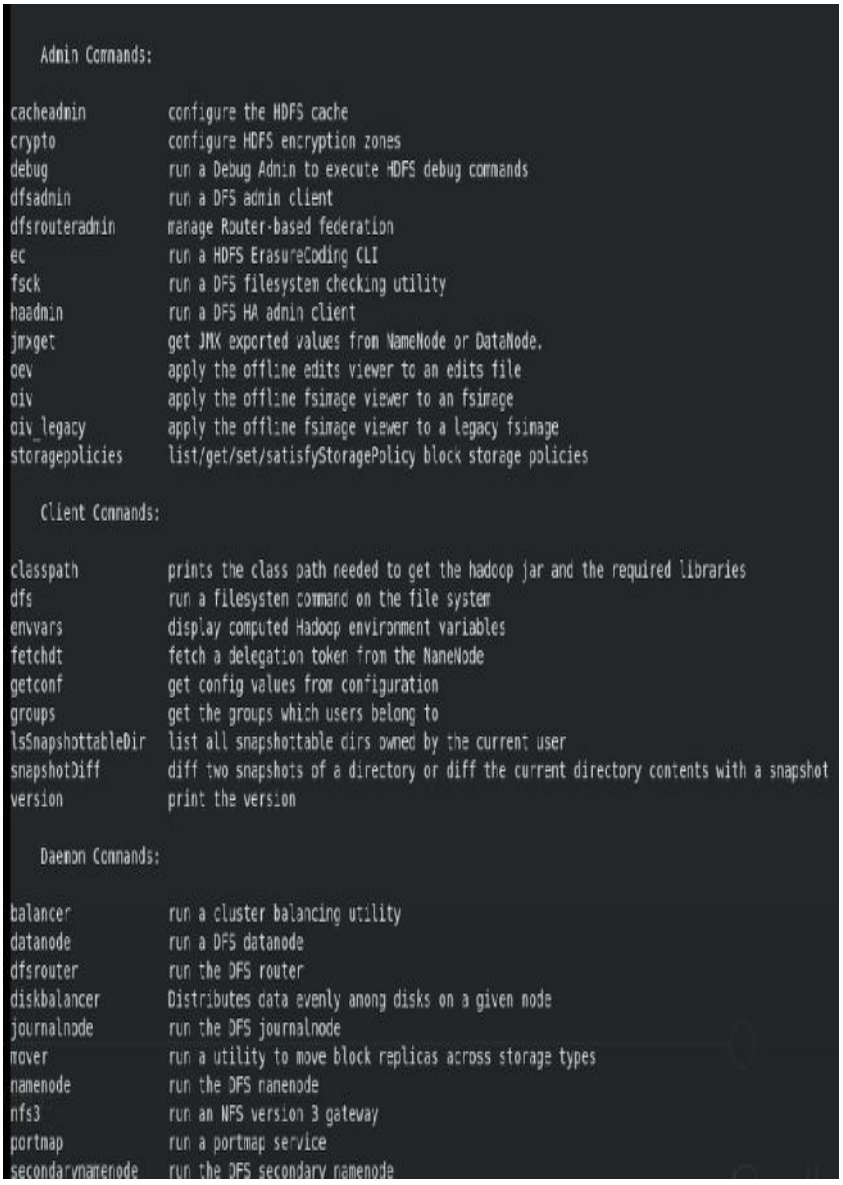


So, we will get a message that namenode has been successfully formatted.

Step 19: To start all the services or start Hadoop daemons. To start services, we will go to sbin folder and will see all services.

start dfs.sh


[rajni@localhost sbin]$ ls
distribute-exclude.sh
FederationStateStore
hadoop-daemon.sh
hadoop-daemons.sh
hdfs-config.cmd
hdfs-config.sh
httpfs.sh
kms.sh
mr-jobhistory-daemon.sh
refresh-namenodes.sh
slaves.sh
start-all.cmd
start-all.sh
start-balancer.sh
start-dfs.cmd
start-dfs.sh
start-secure-dns.sh
start-yarn.cmd
start-yarn.sh
stop-all.cmd
stop-all.sh
stop-balancer.sh
stop-dfs.cmd
stop-dfs.sh
stop-secure-dns.sh
stop-yarn.cmd
stop-yarn.sh
yarn-daemon.sh
yarn-daemons.sh

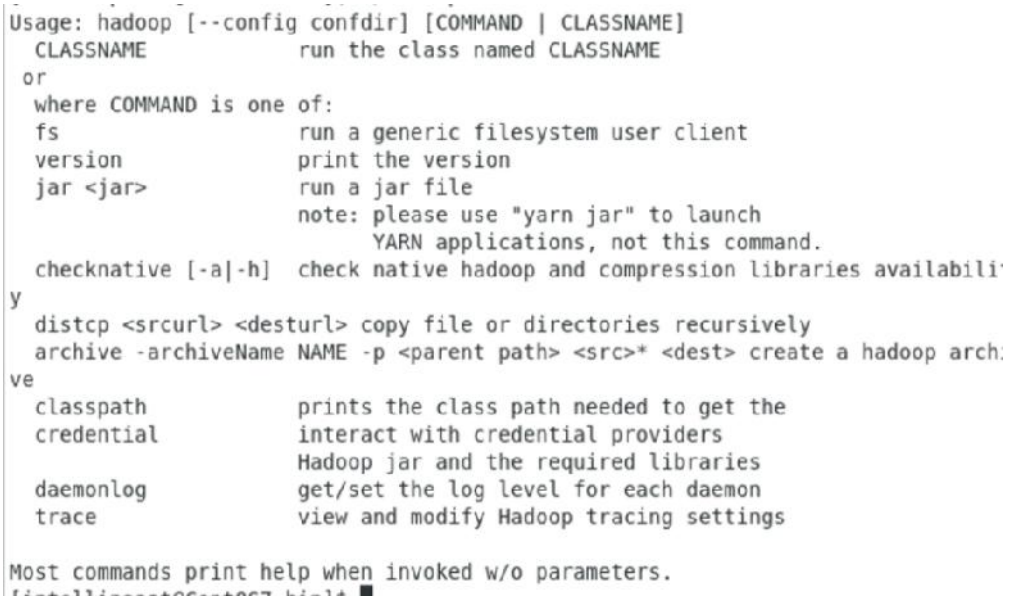

Admin Commands:
cacheadmin configure the HDFS cache
crypto configure HDFS encryption zones
debug run a Debug Admin to execute HDFS debug commands
dfsadmin run a DFS admin client
dfsrouteradmin manage Router-based federation
ec run a HDFS ErasureCoding CLI
fsck run a DFS filesystem checking utility
haadmin run a DFS HA admin client
jmxget get JMX exported values from NameNode or DataNode.
oev apply the offline edits viewer to an edits file
oiv apply the offline fsimage viewer to an fsimage
oiv_legacy apply the offline fsimage viewer to a legacy fsimage
storagepolicies list/get/set/satisfyStoragePolicy block storage policies
Client Commands:
classpath prints the class path needed to get the hadoop jar and the required libraries
dfs run a filesystem command on the file system
envvars display computed Hadoop environment variables
fetchdt fetch a delegation token from the NameNode
getconf get config values from configuration
groups get the groups which users belong to
lsSnapshottableDir list all snapshottable dirs owned by the current user
snapshotDiff diff two snapshots of a directory or diff the current directory contents with a snapshot
version print the version
Daemon Commands:
balancer run a cluster balancing utility
datanode run a DFS datanode
dfsrouter run the DFS router
diskbalancer Distributes data evenly among disks on a given node
journalnode run the DFS journalnode
mover run a utility to move block replicas across storage types
namenode run the DFS namenode
nfs3 run an NFS version 3 gateway
portmap run a portmap service
secondarynamenode run the DFS secondary namenode

Step20:Checking Hadoop

You must now verify whether the Hadoop installation was completed successfully on your machine.

Go to the directory where you extracted the Hadoop tar file, right-click on the bin, and select Open in Terminal from the menu.

Now type the command ls. If you get a window like the one below, that means Hadoop has been successfully installed!



6.1 HDFS

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly faulted tolerant and designed using low-cost hardware.HDFS holds a very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored redundantly to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

- It is suitable for distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of the namenode and datanode help users to easily check the status of the cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

6.2 HDFS Architecture

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are several DataNodes, usually one per node in the cluster, which manages storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The

DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

- HDFS follows the master-slave architecture

Namenode

The NameNode is the centerpiece of an HDFS file system. It keeps thedirectory tree of all files in the file system, and tracks where across the cluster the file data iskept. It does not store the data of these files itself.Clientapplications talk to the Name Node whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The Name Node respondsthe successful requests by returning a list of relevant Data Node servers wherethe data lives.The Name Node is a Single Point of Failure for the HDFS Cluster.HDFS is not currently a High Availability system.When the Name Node goesdown, the file system goes offline. There is an optional Secondary Name The node can be hosted on a separate machine.It only creates checkpoints of thenamespace by merging the edits file into the fsimage file and does not provide anyreal redundancy. Hadoop 0.21+ has a Backup Name Node that is part of a plan to have an HA name service, but it needs active contributions from thepeople who want it (i.e. you) to make it Highly Available.Tracks where acrossthe cluster the file data is kept. It does not store the data of these files itself. Client applications talk to the Name Node . Name Node responds the successful requests. **Name Node** works as Master in Hadoop cluster. Below listed are the main function performed by Name Node:

1. Stores metadata of actual data.
2. Manages File system namespace.

Regulates client access request for actual file data file. Assign work toSlaves(DataNode). Executes file system name space operation like opening/closing files, renaming files and directories.As Name node keep metadata in memory for fast retrieval, the huge amount of memory is required for its operation. This should be hosted on reliable hardware.

Data node

Data Node works as Slave in Hadoop cluster . Below listed are the main function performed by Data Node:

1. Actually stores Business data.
2. This is actual worker node were Read/Write/Data processing is handled.

Upon instruction from Master, it performs creation/replication/deletion of data blocks. As all the Business data is stored on Data Node, the huge amount of storage is required for its operation. Commodity hardware can be used for hosting Data Node.Data Node is responsible for storing the actual data in HDFS. Data Node is also known as the Slave. Name Node and data Node are in constant communication.DataNode is responsible for storing the actual data in HDFS. DataNode is also known as the Slave. NameNode and DataNode are in constant communication.When a Data Node is down, it does not affect the availability of data or the cluster. Name Node will arrange for replication for the blocks managed by the Data Node that is not available.DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode. **DataNode periodically send HEARTBEATS to NameNode.** DataNode is responsible for storing the actual data in HDFS. DataNode is also known as the Slave. NameNode and DataNode are in constant communication.

Block

Generally, the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goal of HDFS

- **Fault detection and recovery** − Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.
- **Huge datasets** − HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

- **Hardware at data** − A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

6.3 Commands in HDFS

jps

This command is a part of Java since v1.5.0.jps stands for Java Virtual Machine Process Status Tool.In our practice, when we start the single node cluster following processes are must be up and running:

Name Node
Data Node
Resource Manager
Node Manager

jps is a tool to check, whether expected Hadoop processes are up and in running state or not.

- The syntax is

Jps

- **ls**

Using the ls command, we can check for the directories in HDFS.

List the HDFS contents. Hadoop Prefix

Every command in Hadoop have prefix:

hadoop fs

Or

hdfsdfs

List the Contents that are in Hadoop

hadoop fs -ls

Display the Contents of Directory

- Syntax

$hadoop fs –ls directoryname

Create a Directory in hdfs

- Syntax

hadoop fs –mkdirabc

Verify a Directory in hdfs

- Syntax

hadoop fs –ls

Every command start with (-)hyphen in hdfs.

Syntax to Create a File

$hadoop fs –touchz file1.txt

This command will create an empty file in home location.

Create a File of Zero Length.

Copy from the local file system into hdfs directory

Firstly create a simple file.

gedit f1.txt

$hadoop fs –copyFromLocal f1.txt file1

Or

put(Copy single src, or multiple srcs from local file system to the destination file system).

Create a File

To check either f1.txt is available or not. To check the contents of the file.

$hadoop fs –cat abc/f1.txt

Sending a File from hdfs to Local File System

Syntax

Hadoop fs –copyToLocalabc/f1.txt ~/desktop/

Verify either it is available in desktop location or not:

cd Desktop/

ls

Verify either it's the same File

cat f1.txt

Copy Command

- Sending file from hdfs to hdfs directory
- Syntax is

hadoop fs –mkdir abc1

Hadoop fs –cp abc/f1.txt abc1/

–cp abc/f1.txt - Source

abc1/ - Destination

Verify

$hadoop –fs –cat abc1/f1.txt

Move the Directory from one hdfs to other hdfs

Firstly, create another directory with abc2

hadoop fs –mkdir abc2

Hadoop fs –mv abc/f1.txt abc2/

Check the Content with cat Command

hadoop fs –cat abc2/f1.txt

Which Directory is Taking More Space?

hadoop fs –du abc1

-du	-dus
Display size of each and every file	Display total size

Test Command: To Check Some Conditions

- Syntax is

$hadoop fs –test d/e/f/w/z

$Hadoop fs –test –d destination

To print result

Echo $?

0 means it's a directory

1 means it's a file

To Check Whether Path Exist or Not

$Hadoop fs –test –e destination

To print result

Echo $?

0 means it is existing

1 means does not exist

To Check Whether Given Path is File or Not

$hadoop fs –test –f destination

To print result

Echo $?

0 means it is file

1 means it is not file

To Check Either Given File have Some Content or its Empty File

$hadoop fs –test –z destination

To print result

Echo $?

0 means it is zero content file

1 means it is non-zero content file.

Move a File from Local File System to hdfs System

Syntax

$hadoop fs –moveFromLocal a2.txt destination/

Verify

cat a2.txt

It will return error because file has already move to hdfs directory named destination.

getmerge: I want to merge contents of multiple files in one file which will be available under the hdfs.

- Syntax

$Hadoop fs –cat destination/xyz.txt(to check contents of file)

$Hadoop fs –cat destination/xyz1.txt

$hadoop fs –getmerge –nl destination/xyz.txt destination/xyz1.txt ~/Destop.mergeresult.txt

Check with cat command

$cat Desktop/mergeresult.txt

Append to File: Append the Contents of Two Files(in Local System) Which is available under the hdfs

$gedit q1.txt

$gedit q2.txt

$hadoop fs –touchz q3.txt

$hadoop fs –appendToFile q1.txt q2.txt q3.txt

Q3 will be having data of both the files.

Checksum Command: to Verify the Integrity of the File. Whether File is Modifies or Not.

- Syntax

$hadoop fs –checksum destination/xyz.txt

Fsck: Check Status of Particular Location

To check for root

$hadoopfsck - /

$hadoop fs –rm destination/xyz.txt

To Change Group of the File

- Syntax

$hadoop fs –ls destination/

Hadoop fs –chgrpamrit destination/a2.txt

Stat Command: To Check Stat

$hadoop fs –stat %b destination/xyz1.txt

To Check Supername

$hadoop fs –stat %g destination/xyz1.txt

To Check Replication

$hadoop fs –stat %r destination/xyz1.txt

To Check User of this File

%u

To Check Who modified the File Last Time

%y

Summary

- Apache Hadoop is a Java-based open-source software framework for managing data processing and storage in large data applications. Hadoop works by breaking down huge data sets and analytical jobs into smaller workloads that can be handled in parallel across nodes in a computing cluster. Hadoop can handle both organised and unstructured data, and it can scale up from a single server to thousands of servers with ease.
- Java is an object-oriented programming language with a high level of abstraction and as few implementation dependencies as feasible.
- Ssh-keygen is an utility that allows you to generate fresh SSH authentication key pairs. This type of key pair is used to automate logins, provide single sign-on, and authenticate hosts.
- The GNOME desktop environment's official text editor is gedit. Gedit is a strong general-purpose text editor that aims for simplicity and ease of use. It can produce and modify a wide range of text files.
- A bashrc file is a shell script file that Linux uses to load modules and aliases into your profile when it boots up. In your /home directory, you'll find your bashrc file. With a text editor like nano, you can make modifications to it. Adding parts to your bashrc file, such as modules to load when you sign in or aliases for commonly used commands, can help you save time in your workflows.

- The Hadoop daemon receives information from the core-site.xml file about where NameNode is located in the cluster. It provides Hadoop Core configuration parameters, such as I/O settings shared by HDFS and MapReduce.
- The configuration parameters for HDFS daemons; the NameNode, Secondary NameNode, and DataNodes, are included in the hdfs-site. xml file.... xml to provide default block replication and permission checking on HDFS. When the file is generated, the number of replications can also be selected.
- YARN has a resource model that may be expanded. YARN records CPU and RAM for all nodes, applications, and queues by default, but the resource definition may be expanded to include any "countable" resource.
- MapReduce is well suited to iterative computations with massive amounts of data that require parallel processing. Rather than a method, it depicts a data flow. MapReduce may be used to process a graph in parallel. The map, shuffle, and reduce stages of graph algorithms all follow the same pattern.
- An HDFS file system is built around the NameNode. It maintains the directory tree of all files in the file system and records where the file data is stored across the cluster.... In response to successful queries, the NameNode returns a list of relevant DataNode servers where the data is stored.

Keywords

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

Java is a platform as well as a programming language. Java is a high-level programming language that is also robust, object-oriented, and secure.

Process is what daemons stand for. **Hadoop Daemons** are a collection of Hadoop processes. Because Hadoop is a Java platform, all of these processes are Java Processes.

NameNode is a component of the Master System. Namenode's main function is to manage all of the MetaData. The list of files saved in HDFS is known as metadata (Hadoop Distributed File System). In a Hadoop cluster, data is stored in the form of blocks, as we all know.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Data node: The data node is a commodity computer with the GNU/Linux operating system and data node software installed. In a cluster, there will be a data node for each node (common hardware/system).

Map-red: It is one of the most significant configuration files for Hadoop's runtime environment settings. It includes MapReduce's setup options. By setting the MapReduce.framework.name variable in this file, we may give MapReduce a name.

Data dependability refers to the completeness and accuracy of data, and it is a critical basis for establishing data confidence within an organisation. One of the key goals of data integrity programmes, which are also used to maintain data security, data quality, and regulatory compliance, is to ensure data dependability.

Fault tolerance: Because it replicates data across several DataNodes, HDFS is fault-tolerant. A block of data is duplicated on three DataNodes by default. Different DataNodes are used to hold the data blocks. Data can still be obtained from other DataNodes if one node fails.

HBase: HBase is a Hadoop-based open-source database with sorted map data. It's horizontally scalable and column-oriented.

Blocks: Large files were broken into little segments known as Blocks in Hadoop HDFS. The physical representation of data is called a block. Except for the final block, which might be the same size or

less, all HDFS blocks are the same size. Hadoop divides files into 128 MB blocks before storing them in the Hadoop file system.

Self Assessment

1. _____ is the main prerequisite for Hadoop.

A. Java

B. HTML

C. C#

D. None of above

2. On _____ node job tracker runs.

A. Datanode

B. Namenode

C. Secondary namenode

D. Secondary datanode

3. Which command used to verify the existence of java in your system?

A. $java +version

B. $java @version

C. $java -version

D. $java =version

4. Select Java SE package.

A. JRE

B. JDK

C. Both

D. None of above

5. Command to create Hadoop user

A. useradd username

B. hadoopadd username

C. addhadoop username

D. None of above

6. Hadoop cluster operate in three supported modes. Those modes are _____

A. Local/Standalone mode

B. Psuedo Distributed mode

C. Fully Distributed mode

D. All of above

7. Select the command to starts all Hadoop DFS daemons.

A. all-run.sh

B. start-all.sh

C. start-dfs.sh

D. run-dfs.sh

8. Select the command to stop all Hadoop DFS daemons.

A. stop-all.sh

B. all-stop.sh

C. hold-all.sh

D. abort-all.sh

9. JAVA_HOME is set in _____

A. hadoop-env.sh

B. hadoop-environment.sh

C. env-hadoop.sh

D. None of above

10. On hadoop-env.sh, which of the following properties is configured

A. Replication factor

B. Directory names to store hdfs files

C. Host and port where MapReduce task runs

D. Java Environment variables.

11. When a computer is designated as a datanode, the disc space available to it is reduced.

A. Can be used only for HDFS storage

B. Can be used for both HDFS and non-HDFs storage

C. Cannot be accessed by non-hadoop commands

D. Cannot store text files.

12. HDFS stands for

A. Hadoop File Search

B. Hadoop File Switch

C. Hadoop File System

D. Hadoop Field System

13. HDFS operates in a _____ manner.

A. Master-worker fashion

B. Worker-master

C. Master-slave

D. Slave-master fashion

14. Select goals of HDFS

A. Fault detection and recovery

B. Data duplication

C. Data redundancy

D. All of above

15. When the Primary Name Node fails, the _____ Name Node is utilized.

A. Data

B. Primary

C. Secondary

D. None of above

Answers for Self Assessment

1.	A	2.	B	3.	C	4.	C	5.	A
6.	D	7.	B	8.	A	9.	A	10.	D
11.	B	12.	C	13.	C	14.	B	15.	B

Review Questions

1. Explain architecture of Hdfs.
2. Explain all goals of HDFS.
3. Write down HDFS components.
4. What do you understand by configurations file like core-site.xml, yarn-site.xml, hdfs-site.xml.
5. What is function of .ssh keygen

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley. White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/

2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NOSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 07: Hadoop Architecture

CONTENTS

Objectives

- Learn what is hadoop
- Understand the Hadoop Core components
- Learn How Hdfs Works.
- What is Hadoop Cluster
- Learn Architecture of Hadoop Cluster
- HDFS Architecture and Hadoop features

Introduction

Apache Hadoop is an open-source software framework that stores data in a distributed manner and process that data in parallel. Hadoop provides the world's most reliable storage layer – **HDFS**, a batch processing engine – **MapReduce** and a resource management layer – **YARN**.

Figure 1: Introduction to Apache Hadoop

7.1 What is Hadoop Distribute File System (HDFS)

It is difficult to maintain huge volumes of data in a single machine. Therefore, it becomes necessary to break down the data into smaller chunks and store it on multiple machines.File systems that manage the storage across a network of machines are called distributed file systems.Hadoop Distributed File System (HDFS) is the storage component of Hadoop. All data stored on Hadoop is stored in a distributed manner across a cluster of machines. But it has a few properties that define its existence.Hadoop Distributed File System (HDFS) is the storage component of Hadoop. All data stored on Hadoop is stored in a distributed manner across a cluster of machines. But it has a few properties that define its existence.

- **Huge Volumes –** Being a distributed file system, it is highly capable of storing petabytes of data without any glitches.
- **Data Access –** It is based on the philosophy that "the most effective data processing pattern is write-once, the read-many-times pattern".
- **Cost-effective –** HDFS runs on a cluster of commodity hardware. These are inexpensive machines that can be bought from any vendor.

You can't understand the working of Hadoop without knowing its core components. So, Hadoop consists of three layers (core components) and they are: -

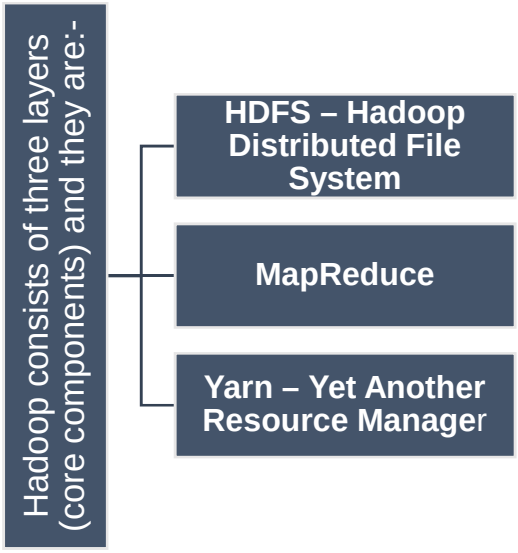
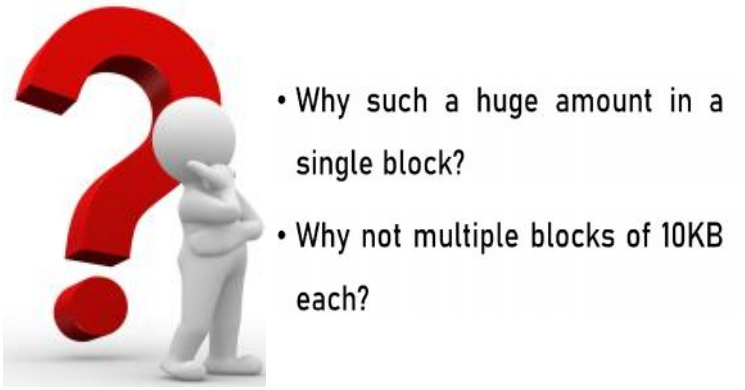


Figure 2: Hadoop Components and Domains

- ***HDFS – Hadoop Distributed File System*** provides for the storage of Hadoop. As the name suggests it stores the data in a distributed manner. The file gets divided into a number of blocks which spreads across the cluster of commodity hardware.This, however, is transparent to the user working on HDFS. To them, it seems like storing all the data onto a single machine.These smaller units are the **blocks** in HDFS. The size of each of these blocks is 128MB by default, you can easily change it according to requirement. So, if you had a fileof size 512MB, it would be divided into 4 blocks storing 128MB each.If, however, you had a file of size 524MB, then, it would be divided into 5 blocks. 4 of these would store 128MB each, amounting to 512MB. And the 5th would store the remaining 12MB. That's right! This last block won't take up the complete 128MB on the disk.
-



Well, the amount of data with which we generally deal with in Hadoop is usually in the order of petra bytes or higher.Therefore, if we create blocks of small size, we would end up with a colossal number of blocks. This would mean we would have to deal with equally large metadata regarding the location of the blocks which would just create a lot of overhead. And we don't really want that!.The file itself would be too large to store on any single disk alone. Therefore, it is prudent to spread it across different machines on the cluster.It would also enable a proper spread of the workload and prevent the choke of a single machine by taking advantage of parallelism.

7.2 Hadoop Components

The HDFS comprises the following components

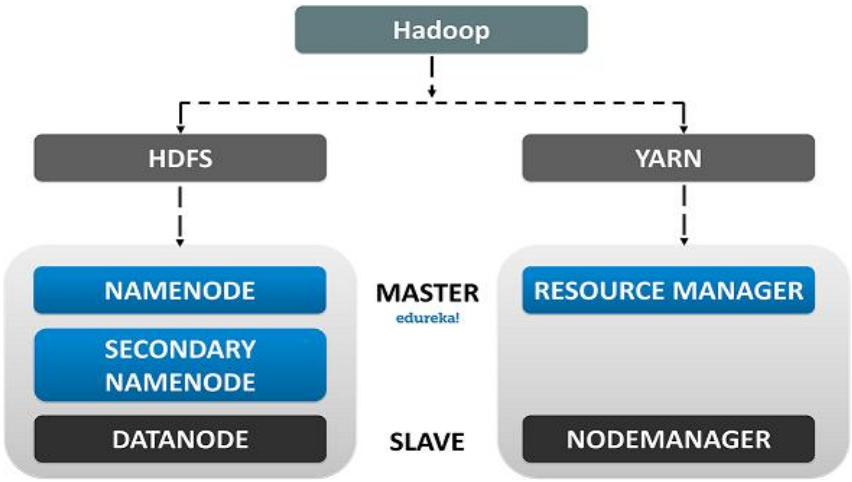


Figure 3: HDFS Components

HDFS operates in a master-slave architecture, this means that there are one master node and several slave nodes in the cluster. The master node is the **Namenode**.

- **Namenode** is the master node that runs on a separate node in the cluster.Manages the filesystem namespace which is the filesystem tree or hierarchy of the files and directories.Stores information like owners of files, file permissions, etc for all the files. It is also aware of the locations of all the blocks of a file and their size.

Namenode in HDFS

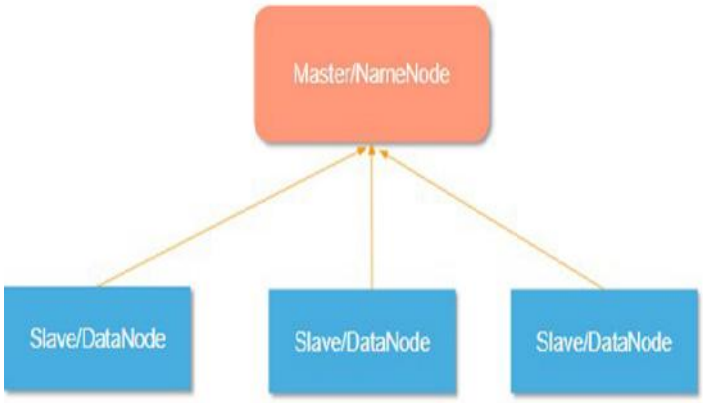
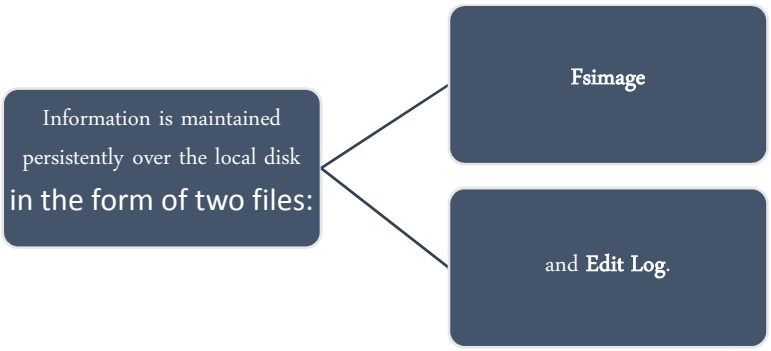


Figure 4: Namenode in HDFS

All this information is maintained persistently over the local disk in the form of two files: **Fsimage** and **Edit Log**.



- **Fsimage** stores the information about the files and directories in the filesystem. For files, it stores the replication level, modification and access times, access permissions, blocks the file is made up of, and their sizes. For directories, it stores the modification time and permissions.
- **Edit Log** on the other hand keeps track of all the write operations that the client performs. This is regularly updated to the in-memory metadata to serve the read requests.

Whenever a client wants to write information to HDFS or read information from HDFS, it connects with the **Namenode**. The Namenode returns the location of the blocks to the client and the operation is carried out.Yes, that's right, the Namenode does not store the blocks. For that, we have separate nodes.**Datanodes** are the worker nodes. They are inexpensive commodity hardware that can be easily added to the cluster.They periodically send heartbeats to the Namenode so that it is aware of their health. With that, a DataNode also sends a list of blocks that are stored on it so that the Namenode can maintain the mapping of blocks to Datanodes in its memory. But in addition to these two types of nodes in the cluster, there is also another node called the Secondary Namenode.**Datanodes** are responsible for storing, retrieving, replicating, deletion, etc. of blocks when asked by the Namenode.They periodically send heartbeats to the Namenode so that it is aware of their health. With that, a DataNode also sends a list of blocks that are stored on it so that the Namenode can maintain the mapping of blocks to Datanodes in its memory.But in addition to these two types of nodes in the cluster, there is also another node called the Secondary Namenode.

Suppose we need to restart the **Namenode**, which can happen in case of a failure. This would mean that we have to copy the Fsimage from disk to memory. Also, we would also have to copy the latest copy of Edit Log to Fsimage to keep track of all the transactions. But if we restart the node after a long time, then the Edit log could have grown in size. This would mean that it would take a lot of

time to apply the transactions from the Edit log. And during this time, the filesystem would be offline. Therefore, to solve this problem, we bring in the **Secondary Namenode**.**Secondary Namenode** is another node present in the cluster whose main task is to regularly merge the Edit log with the Fsimage and produce check- points of the primary's in-memory file system metadata. This is also referred to as **Checkpointing**.

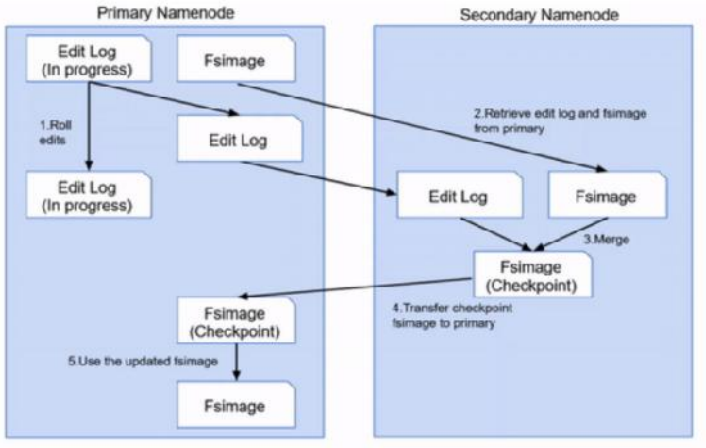


Figure 5: Primary namenode and Secondary namenode

Secondary Namenode in HDFS

- But the checkpointing procedure is computationally very expensive and requires a lot of memory, which is why the Secondary namenode runs on a separate node on the cluster.However, despite its name, the Secondary Namenode does not act as a Namenode. It is merely there for Checkpointing and keeping a copy of the latest Fsimage.
- **Mapreduce**: It is the processing layer in Hadoop. Hadoop MapReduce processes the data stored in Hadoop HDFS in parallel across various nodes in the cluster. It divides the task submitted by the user into the independent task and processes them as subtasks across the commodity hardware.Hadoop MapReduce is the processing unit of Hadoop. In the MapReduce approach, the processing is done at the slave nodes, and the final result is sent to the master node.A data containing code is used to process the entire data. This coded data is usually very small in comparison to the data itself. You only need to send a few kilobytes worth of code to perform a heavy-duty process on computers.

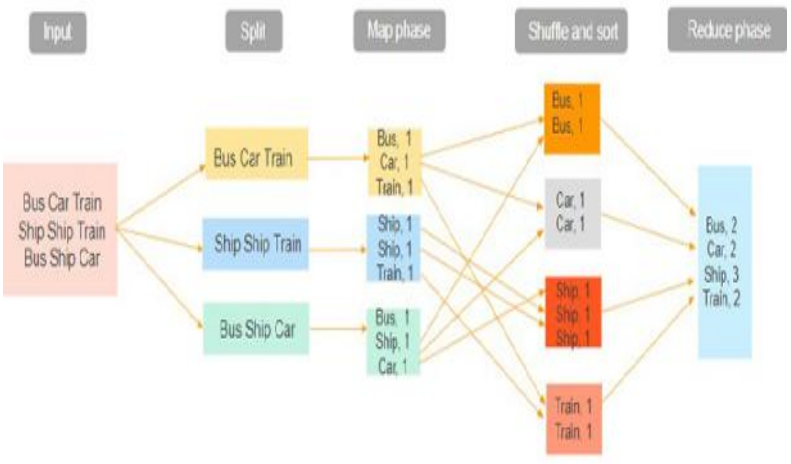


Figure 6: MapReduce

The input dataset is first split into chunks of data. In this example, the input has three lines of text with three separate entities - "bus car train," "ship ship train," "bus ship car." The dataset is then split into three chunks, based on these entities, and processed parallelly.In the map phase, the data is assigned a key and a value of 1. In this case, we have one bus, one car, one ship, and one train.These key-value pairs are then shuffled and sorted together based on their keys. At the reduce phase, the aggregation takes place, and the final output is obtained.

Hadoop YARN

Hadoop YARN stands for Yet Another Resource Negotiator. It is the resource management unit of Hadoop and is available as a component of Hadoop version 2. Hadoop YARN acts like an OS to Hadoop. It is a file system that is built on top of HDFS.It is responsible for managing cluster resources to make sure you don't overload one machine. It performs job scheduling to make sure that the jobs are scheduled in the right place.

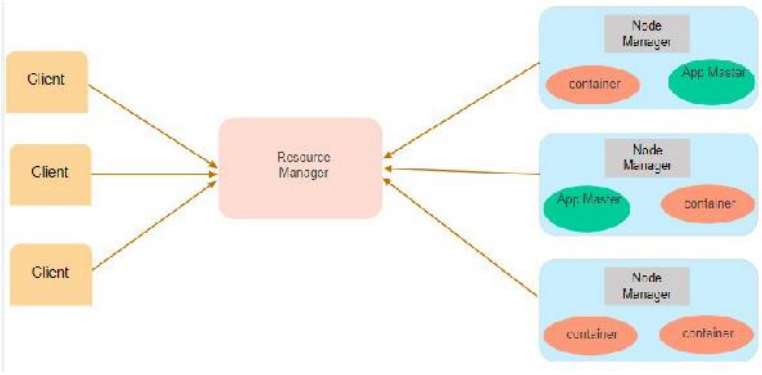


Figure 7 Hadoop YARN

Suppose a client machine wants to do a query or fetch some code for data analysis. This job request goes to the resource manager (Hadoop Yarn), which is responsible for resource allocation and management.In the node section, each of the nodes has its node managers. These node managers manage the nodes and monitor the resource usage in the node. The containers contain a collection of physical resources, which could be RAM, CPU, or hard drives. Whenever a job request comes in, the app master requests the container from the node manager. Once the node manager gets the resource, it goes back to the Resource Manager.

Hadoop Daemons

The Hadoop Daemons arethe processes that run in the background. These 4 daemons run for Hadoop to be functional.

The Hadoop Daemons are:

a) **Namenode –** It runs on master node for HDFS.

b) **Datanode –** It runs on slave nodes for HDFS.

c) **Resource Manager –** It runs on **YARN** master node for MapReduce.

d) **Node Manager –** It runs on YARN slave node for MapReduce.

Whenever the client wants to perform any processing on its data in the Hadoop cluster, then it first stores the data in Hadoop HDFS and then writes the MapReduce program for processing the Data. The Hadoop MapReduce works as follows:

1. Hadoop divides the job into tasks of two types, that is, map tasks and reduce tasks. YARN scheduled these tasks (which we will see later in this article). These tasks run on different DataNodes.

2. The input to the MapReduce job is divided into fixed-size pieces called input splits.

3. One map task which runs a user-defined map function for each record in the input split is created for each input split. These map tasks run on the DataNodes where the input data resides.

4. The output of the map task is intermediate output and is written to the local disk.

5. The intermediate outputs of the map tasks are shuffled and sorted and are then passed to the reducer.6. For a single reduce task, the sorted intermediate output of mapper is passed to the node where the reducer task is running. These outputs are then merged and then passed to the user-defined reduce function.

7. The reduce function summarizes the output of the mapper and generates the output. The output of the reducer is stored on HDFS.

8. For multiple reduce functions, the user specifies the number of reducers. When there are multiple reduce tasks, the map tasks partition their output, creating one partition for each reduce task.

How HDFS Works?

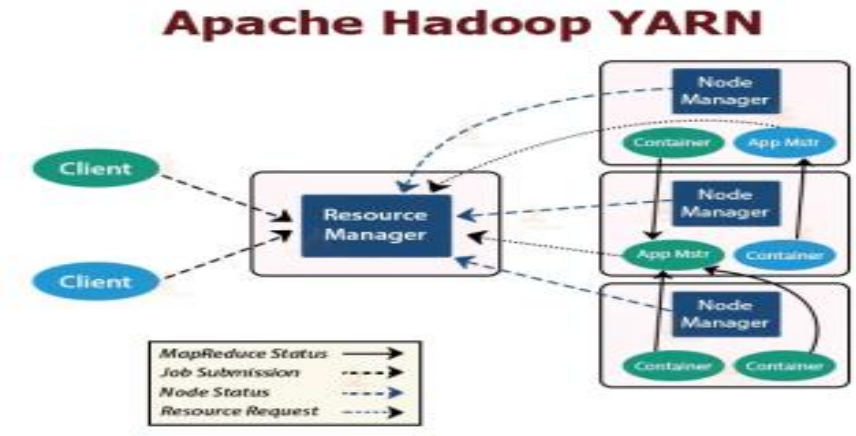


Figure 8: Apache Hadoop YARN

YARN is the resource management layer in Hadoop. It schedules the task in the Hadoop cluster and assigns resources to the applications running in the cluster. It is responsible for providing the computational resources needed for executing the applications.There are two YARN daemons running in the Hadoop cluster for serving YARN core services. They are:

a. **ResourceManager:** It is the master daemon of YARN. It runs on the master node per cluster to manage the resources across the cluster. The ResourceManager has two major components that are **Scheduler** and **ApplicationManager.**

- The **scheduler** allocates resources to various applications running in the cluster.
- **ApplicationManager** takes up the job submitted by the client, and negotiates thecontainer for executing the application-specific ApplicationMaster, and restarts the ApplicationMaster container on failure.

b. **Node Manager:** NodeManager is the slave daemons of YARN. It runs on all the slave nodes in the cluster. It is responsible for launching and managing the containers on nodes. Containers execute the application-specific processes with a constrained set of resources such as memory, CPU, and so on.When NodeManager starts, it announces himself to the ResourceManager. It periodically sends a heartbeat to the ResourceManager. It offers resources to the cluster.

c. **Application Master:** The per-application ApplicationMaster negotiates containers form schedulers and tracks container status and monitors the container progress.A client submits an application to the ResourceManager. The ResourceManager contacts the NodeManager that launches and monitors the compute containers on nodes in the cluster. The container executes the ApplicationMaster.

The MapReduce task and the ApplicationMaster run in containers which are scheduled by the ResourceManager and managed by the NodeManagers.

7.3 Summarize How Hadoop Works Internally

1. HDFS divides the client input data into blocks of size 128 MB. Depending on the replication factor, replicas of blocks are created. The blocks and their replicas are stored on different DataNodes.
2. Once all blocks are stored on **HDFS DataNodes**, the user can process the data.
3. To process the data, the client submits the **MapReduce** program to Hadoop.
4. **ResourceManager** then scheduled the program submitted by the user on individual nodes in the cluster.
5. Once all nodes complete processing, the output is written back to the HDFS.

7.4 Hadoop Cluster

A Hadoop cluster is nothing but a group of computers connected together via LAN. We use it for storing and processing large data sets. Hadoop clusters have a number of commodity hardware connected together. They communicate with a high-end machine which acts as a master. These master and slaves implement distributed computing over distributed data storage. It runs open-source software for providing distributed functionality.

Architecture of Hadoop

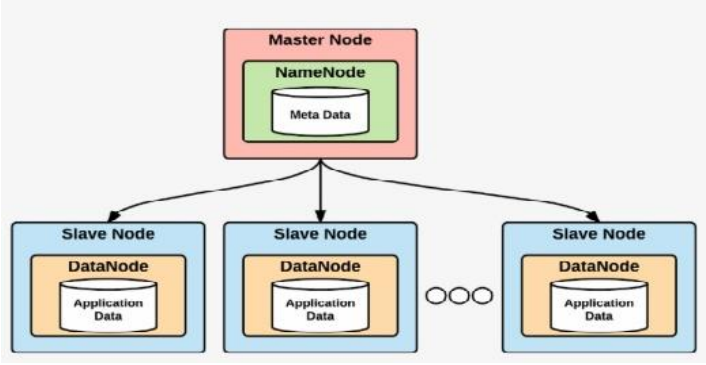


Figure 9: Master/Slave Architecture

It is a machine with a good configuration of memory and CPU. There are two daemons running on the master and they are Name Node and Resource Manager.

i. Functions of Name Node

Manages file system namespace, Regulates access to files by clients, Stores metadata of actual data for example – file path, number of blocks, block id, the location of blocks etc.

Executes file system namespace operations like opening, closing, renaming files and directories, The Name Node stores the metadata in the memory for fast retrieval. Hence, we should configure it on a high-end machine.

ii. Functions of Resource Manager

It arbitrates resources among competing nodes, Keeps track of live and dead nodes

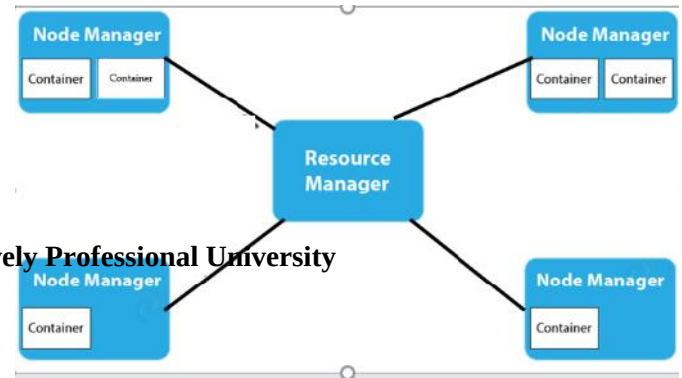


Figure 10: Function of Resource Manager

Slaves in the Hadoop Cluster

It is a machine with a normal configuration. There are two daemons running on Slave machines and they are – DataNode and Node Manager

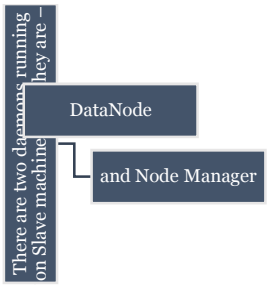


Figure 11: Slaves in the Hadoop Cluster

Functions of a Data Node

- It stores the business data, It does read, write and data processing operations, Upon instruction from a master, it does creation, deletion, and replication of data blocks.

Functions of a Node Manager

- It runs services on the node to check its health and reports the same to Resource Manager. We can easily scale Hadoop cluster by adding more nodes to it. Hence, we call it a linearly scaled cluster.
- Each node added increases the throughput of the cluster. We **install Hadoop** and configure it on client nodes

Functions of the Client Node

- To load the data on the Hadoop cluster, tells how to process the data by submitting MapReduce job., Collects the output from a specified location.

7.5 What is Hadoop High Availability?

With Hadoop 2.0, we have support for multiple Name Nodes and with Hadoop 3.0 we have standby nodes. This overcomes the SPOF (Single Point Of Failure) issue using an extra Name Node (Passive Standby Name Node) for automatic failover. This is the high availability in Hadoop.

Figure 12 Single Point of Failure

What is Failover?

Failover is a process in which the system transfers control to a secondary system in an event of failure.

- **Graceful Failover –** In this type of failover the administrator manually initiates it. We use graceful failover in case of routine system maintenance. There is a need to manually transfer the control to standby NameNode it does not happen automatically.
- **Automatic Failover –** In **Automatic Failover**, the system automatically transfers the control to standby NameNode without manual intervention. Without this automatic failover if the NameNode goes down then the entire system goes down.

Hence the feature of Hadoop high availability is available only with this automatic failover, it acts as your insurance policy against a single point of failure.

7.6 HDFS Architecture

This architecture gives you a complete picture of the Hadoop Distributed File System. There is a single NameNode that stores metadata, and there are multiple DataNodes that do actual storage work. Nodes are arranged in racks, and replicas of data blocks are stored on different racks in the cluster to provide fault tolerance.In the remaining section of this tutorial, we will see how read and write operations are performed in HDFS? To read or write a file in HDFS, the client needs to interact with NameNode. HDFS applications need a write-once-read-many access model for files. A file, once created and written, cannot be edited.NameNode stores metadata, and DataNode stores actual data. The client interacts with NameNode for performing any tasks, as NameNode is the centerpiece in the cluster.There are several DataNodes in the cluster which store HDFS data in the local disk. DataNode sends a heartbeat message to NameNode periodically to indicate that it is alive. Also, it replicates data to other DataNode as per the replication factor.

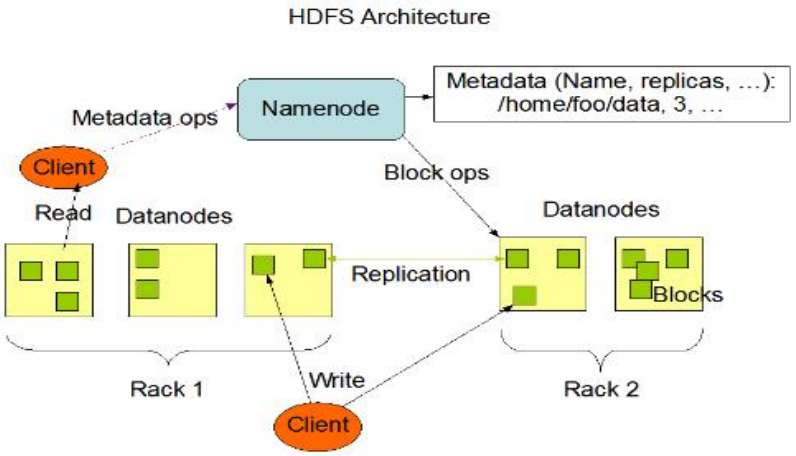


Figure 13: HDFS Architecture

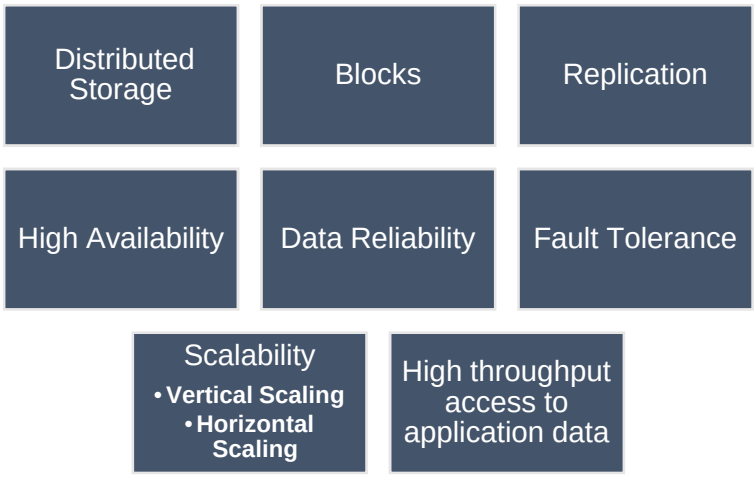


Figure 14: Hadoop HDFS Features

Distributed Storage

HDFS stores data in a distributed manner. It divides the data into small pieces and stores it on different Data Nodes in the cluster. In this manner, the Hadoop Distributed File System provides a way to MapReduce to process a subset of large data sets broken into blocks, parallelly on several nodes. MapReduce is the heart of Hadoop, but HDFS is the one who provides it all these capabilities.

Blocks

HDFS splits huge files into small chunks known as blocks. Block is the smallest unit of data in a filesystem. We (client and admin) do not have any control on the block like block location. NameNode decides all such things.HDFS default block size is 128 MB. We can increase or decrease the block size as per our need. This is unlike the OS filesystem, where the block size is 4 KB. If the data size is less than the block size of HDFS, then block size will be equal to the data size.For example, if the file size is 129 MB, then 2 blocks will be created for it. One block will be of default size 128 MB, and the other will be 1 MB only and not 128 MB as it will waste the space (here block size is equal to data size). Hadoop is intelligent enough not to waste the rest of 127 MB. So it is allocating 1 MB block only for 1 MB data. The major advantage of storing data in such block size is that it saves disk seek time and another advantage is in the case of processing as mapper processes 1 block at a time. So 1 mapper processes large data at a time.

Replication

Hadoop HDFS creates duplicate copies of each block. This is known as replication. All blocks are replicated and stored on different DataNodes across the cluster. It tries to put at least 1 replica in a different rack.

High Availability

Replication of data blocks and storing them on multiple nodes across the cluster provides high availability of data. As seen earlier in this Hadoop HDFS tutorial, the default replication factor is 3, and we can change it to the required values according to the requirement by editing the configuration files (hdfs-site.xml).

Data Reliability

As we have seen in high availability in this HDFS tutorial, data is replicated in HDFS; It is stored reliably as well. Due to replication, blocks are highly available even if some node crashes or some hardware fails. If the DataNode fails, then the block is accessible from other DataNode containing a replica of the block. Also, if the rack goes down, the block is still available on the different rack. This is how data is stored reliably in HDFS and provides fault-tolerant and high availability.

Fault Tolerance

HDFS provides a fault-tolerant storage layer for Hadoop and other components in the ecosystem.HDFS works with commodity hardware (systems with average configurations) that has high chances of getting crashed at any time. Thus, to make the entire system highly fault-tolerant, HDFS replicates and stores data in different places.

Scalability

Scalability means expanding or contracting the cluster. We can scale Hadoop HDFS in 2 ways.

1. **Vertical Scaling:** We can add more disks on nodes of the cluster. For doing this, we need to edit the configuration files and make corresponding entries of newly added disks. Here we need to provide downtime though it is very less. So people generally prefer the second way of scaling, which is horizontal scaling.
2. **Horizontal Scaling:** Another option of scalability is of adding more nodes to the cluster on the fly without any downtime. This is known as horizontal scaling. We can add as many nodes as we want in the cluster on the fly in real-time without any downtime. This is a unique feature provided by Hadoop.

High Throughput Access to Application Data

Hadoop Distributed File System provides high throughput access to application data. Throughput is the amount of work done in a unit time. It describes how fast the data is getting accessed from the system, and it is usually used to measure the performance of the system.In HDFS, when we want to perform a task or an action, then the work is divided and shared among different systems. So all the systems will be executing the tasks assigned to them independently and in parallel. So the work will be completed in a very short period of time. Thus, by reading data in parallel HDFS gives good throughput.

Summary

- Apache Hadoop is a Java-based open-source software framework for managing data processing and storage in large data applications. Hadoop works by breaking down huge data sets and analytical jobs into smaller workloads that can be handled in parallel across nodes in a computing cluster.
- Hadoop applications use the Hadoop Distributed File Solution (HDFS) as their primary data storage system. HDFS is a distributed file system that uses a NameNode and DataNode architecture to allow high-performance data access across highly scalable Hadoop clusters.
- YARN is one of Apache Hadoop's main components, and it's in charge of assigning system resources to the many applications operating in a Hadoop cluster, as well as scheduling jobs to run on different cluster nodes.
- MapReduce is well suited to iterative computations with massive amounts of data that require parallel processing. Rather than a method, it depicts a data flow. MapReduce may be used to process a graph in parallel. The map, shuffle, and reduce stages of graph algorithms all follow the same pattern.
- An HDFS file system is built around the NameNode. It maintains the directory tree of all files in the file system and records where the file data is stored across the cluster.... In response to successful queries, the NameNode returns a list of relevant DataNode servers where the data is stored.

- Hadoop YARN stands for Yet Another Resource Negotiator (YARN). There is a requirement to manage resources at both a global and a node level in a Hadoop cluster.
- In a Hadoop cluster, MapReduce is a programming paradigm that permits tremendous scalability over hundreds or thousands of computers. MapReduce, as the processing component, lies at the heart of Apache Hadoop
- Hadoop Ecosystem is a platform or a suite that offers a variety of services to address big data issues. It consists of Apache projects as well as a variety of commercial tools and solutions. HDFS, MapReduce, YARN, and Hadoop Common are the four core components of Hadoop.

Keywords

- **Hadoop:** Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.
- **Failover:**If the primary system fails or is taken down for maintenance, failover is a backup operational mode that immediately switches to a standby database, server, or network. Failover technology smoothly sends requests from a downed or failing system to a backup system that replicates the operating system environment.
- **HDFS**: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.
- **Name node:** The name node is a piece of commodity hardware that houses the GNU/Linux operating system as well as name node software. It's a piece of software that can run on standard hardware.
- **Data node:** The data node is a commodity computer with the GNU/Linux operating system and data node software installed. In a cluster, there will be a data node for each node (common hardware/system).
- **Data dependability** refers to the completeness and accuracy of data, and it is a critical basis for establishing data confidence within an organisation. One of the key goals of data integrity programmes, which are also used to maintain data security, data quality, and regulatory compliance, is to ensure data dependability.
- **Fault tolerance:** Because it replicates data across several DataNodes, HDFS is fault-tolerant. A block of data is duplicated on three DataNodes by default. Different DataNodes are used to hold the data blocks. Data can still be obtained from other DataNodes if one node fails.
- **HBase:** HBase is a Hadoop-based open-source database with sorted map data. It's horizontally scalable and column-oriented.
- **Blocks:**Large files were broken into little segments known as Blocks in Hadoop HDFS. The physical representation of data is called a block. Except for the final block, which might be the same size or less, all HDFS blocks are the same size. Hadoop divides files into 128 MB blocks before storing them in the Hadoop file system.

Self Assessment

1. Filesystems that manage the storage across a network of machines are called _____

A. Distributed file systems

B. Distributed field systems
C. Distributed file switch
D. None of above

2. Select correct layers of Hadoop.
A. HDFS
B. MapReduce
C. YARN
D. All of the above

3. HDFS operates in a master-slave architecture, this means that there are one master node and several slave nodes in the cluster. The master node is the _____.
A. Datanode
B. Namenode
C. Both
D. All of the above

4. In which of the following files information is maintained persistently over the local disk.
A. Fsimage and Edit log
B. Edit log and Fedit
C. Fsimage and Fedit
D. All of the above

5. Which of the following are Hadoop daemons?
A. Resource Manager
B. Datanode
C. Namenode
D. All of the above

6. In which of the following tasks hadoop divides the job.
A. Map tasks and reduce tasks
B. Reduce tasks and datanode
C. Namenode and datanode
D. None of the above

7. Who developed Hadoop?
A. JetBrains
B. Graydone Hoare
C. Doug Cutting
D. None of the above

8. In Hadoop, which of the following is a distributed data warehouse?
A. Pig
B. Hbase
C. Hive
D. All of the above

9. In Hadoop, which of the following does the Job control?
A. Task class
B. Mapper class
C. Reducer class
D. Job class

10. Slave computers have two daemons operating, and they are
A. Nodemanager and edgenode
B. Edgenode and datanode
C. Factnode and datanode
D. Datanode and node manager

11. Hadoop manages the jobs by breaking them down into _____.
A. Smaller chats.
B. Smaller chunks.
C. Sink chunks.
D. None of the above

12. Failover is a process in which the system transfers control to a secondary system in an event of failure.
A. Graceful Failover
B. Failover
C. Automatic failover
D. All of the above

13. HDFS splits huge files into small chunks known as _____
A. File
B. Blocks
C. Both
D. None of the above

14. Which of the following is HDFS's default block size?
A. 65MB
B. 125MB
C. 128MB
D. 120MB

15. Each block in Hadoop HDFS is duplicated twice. This is referred to as _____.
A. Job tracker
B. Replication
C. Both
D. None of the above

Answers for Self Assessment

1. A 2. D 3. B 4. A 5. D

6.	A	7.	C	8.	C	9.	D	10.	D
11.	B	12.	B	13.	B	14.	C	15.	B

Review Questions

1. Explain architecture of Hadoop.
2. Explain all Hadoop HDFS features.
3. Write down HDFS components.
4. Difference between yarn and MapReduce.
5. Write note on
   A. Data reliability
   B. Replication
   C. Fault tolerance

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
  of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NOSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 08: Hadoop Master Slave Architecture

CONTENTS

Objectives

- Learn Hadoop-MapReduce
- Learn Hadoop – Streaming
- Learn setup of the Hadoop Multi-Node cluster on a distributed environment.
- Learn how to create a system user account

Introduction

Hadoop's MapReduce architecture is used to process massive amounts of data in parallel on enormous clusters of

hardware in a secure way. It allows an application to store data in a distributed format and process large datasets across groups of computers using simple programming models, which is why MapReduce is a programming model for processing large amounts of data distributed across a number of clusters using steps such as input splits, Map, Shuffle, and Reduce.

There are various subprojects within the Apache Hadoop project:

The **Hadoop Common** contains utilities that help the other Hadoop subprojects.

Hadoop Distributed File System (HDFS): Hadoop Distributed File System allows application data to be accessed from a distributed file.

Hadoop MapReduce is a software framework that allows huge distributed data sets to be processed on compute clusters.

Hadoop YARN is a framework for managing resources and scheduling jobs in Hadoop.

8.1 How does Hadoop's MapReduce make working so simple?

MapReduce allows data processing to be scaled across hundreds or thousands of cluster processors. The MapReduce paradigm consists of two processes: map and reduce, as well as processing steps termed mapper and reducer. Scaling up to operate over multiples or even thousands of clusters is just a configuration modification once we've written MapReduce for an application. Many programmers have been drawn to the MapReduce approach because of this capability.

How does Hadoop's MapReduce work?

The MapReduce program executes mainly in Four Steps :

- Input splits
- Map
- Shuffle
- Reduce

Now we will see each step how they work.

The input splits and Map steps are combined in this stage. The source file is transmitted line by line via the Map phase. The input is separated into tiny fixed-size Input splits before it is passed to the Map function task. The input split is a portion of data that a single map may absorb. Each split data is sent to the mapper function in the Map stage, which processes the data before returning results. The job input data for the map or mapper is usually stored in the Hadoop file system as a file or directory (HDFS).

This step is the result of combining the Shuffle and Reduce steps. The reduce function, also known as the Reducer's job, takes the data returned by the map function and processes it by decreasing the role to generate a new set of effects, which are then stored in the HDFS.

A Hadoop framework doesn't know which job each cluster does, either Map or Reduce, or both Map and Reduce. As a result, the Map and Reduce jobs' requests should be routed to the cluster's relevant servers. All of the responsibilities of issuing, confirming job completion, getting data from HDFS, transferring data to the nodes' group, and so on are handled by the Hadoop framework. Hadoop does most of its computing on nodes and stores data on nodes, which lowers network traffic.

As a result, the MapReduce framework complements the Hadoop framework.

Terminology

Payload: The Map and Reduce functions are implemented by PayLoad Applications, which are at the heart of the work.

Mapper: The input key/value pairs are mapped to a collection of intermediate key/value pairs by the Mapper.

Namenode: Node that administers the Hadoop Distributed File System is known as NamedNode (HDFS).

DataNode is a node in which data is delivered in advance of any processing.

JobTrackeroperates on the MasterNode, which takes work requests from clients.

SlaveNode: This is the node where the Map and Reduce programmes are performed.

JobTrackeris a programme that schedules jobs and assigns them to Task Tracker.

Task Tracker keeps track of the task and updates JobTracker on its progress.

Job:A job is a programme that runs a Mapper and Reducer on a dataset.

Task:A Mapper or Reducer task is the execution of a Mapper or Reducer on a slice of data.

Task Attempt is a specific instance of a task execution attempt on a SlaveNode.

Advantages of MapReduce

Scalable: Hadoop is very scalable because to MapReduce, which allows big data sets to be stored in distributed form across numerous servers. Because it is distributed over different servers, it may run in parallel.

Cost-effective solution: MapReduce is a very cost-effective option for organisations that need to store and process large amounts of data in a very cost-effective way, which is a current business requirement.

Flexibility: Hadoop is incredibly adaptable when it comes to multiple data sources and even different types of data, such as structured and unstructured data, thanks to MapReduce. As a result, it gives you a lot of flexibility when it comes to accessing and processing organised and unstructured data.

Fast: Because Hadoop stores data in a distributed file system, which stores data on a cluster's local disc, and MapReduce algorithms are often stored on the same servers, data processing is faster because there is no need to retrieve data from other servers.

Parallel processing: Because Hadoop stores data in a distributed file system and runs a MapReduce algorithm, it separates jobs into map and reduce tasks that may run in parallel. Furthermore, due of the simultaneous execution, the overall run time is reduced.

8.2 Hadoop-Streaming

It is a Hadoop distribution feature that lets developers and programmers to construct Map-Reduce programmes in a variety of programming languages such as Ruby, Perl, Python, C++, and others. Any language that can read from standard input (STDIN), such as keyboard input, and write to standard output can be used (STDOUT). Although the Hadoop Framework is designed entirely in Java, Hadoop apps do not have to be written in the Java programming language. Hadoop Streaming is a functionality that has been available since Hadoop version 0.14.1.

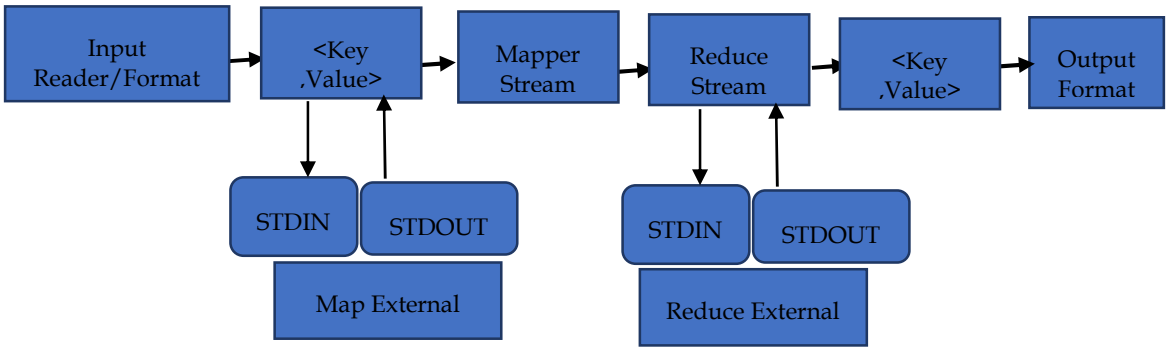
How Hadoop Streaming works?

The flow depicted in a dotted block in the following example graphic is a simple MapReduce job. In that case, we have an Input Reader that reads the input data and generates a list of key-value pairs. We can read data from a database table in.csv format, delimiter format, picture data (.jpg,.png), audio data, and so on. The sole prerequisite for reading all of these sorts of data is that we use these input readers to build a specific input format for that data. The input reader has all of the logic related to the data it is reading. If we wish to read a picture, we must first describe the logic in the input reader in order for it to read the image data and then build key-value pairs for that image data.

We can run whatever language we choose by running it as a separate process. We'll do this by creating an external mapper and running it as a separate process. The standard MapReduce flow does not include these extra map processes. This external mapper will receive STDIN input and output it to STDOUT. As key-value pairs are sent to the internal mapper, the internal mapper process sends them to the external mapper, where we have written our code in another language, such as Python, using STDIN. These external mappers now process these key-value pairs and, via STDOUT, construct intermediate key-value pairs to deliver to the internal mappers.

Reducer, on the other hand, accomplishes the same goal. Once the intermediate key-value pairs have been processed through the shuffle and sorting processes, they are fed to the internal reducer, which sends them to external reducer processes that are working separately via STDIN and gathers the output generated by external reducers via STDOUT, before storing the output to our HDFS.

If we're receiving picture data, we may create a key-value pair for each pixel, with the key being the pixel's position and the value being the colour value (0-255) for a colourful image. This list of key-value pairs is now provided to the Map phase, which works on each of these key-value pairs of each pixel to generate some intermediate key-value pairs, which are then fed to the Reducer after shuffling and sorting, and the Reducer's final output is stored to HDFS. This is how a basic Map-Reduce operation operates.



8.3 nstalling Java

Hadoop requires Java as a prerequisite. To begin, use "java -version" to check for the presence of java on your machine. The java version command's syntax is listed below.

$java -version

8.4 Creating User Account

To use the Hadoop installation, create a system user account on both the master and slave servers.

useradd hadoop2

passwd hadoop2

Mapping the nodes

On all nodes, you must edit the hosts file in the /etc/ folder, specifying the IP address of each system followed by their host names.

vi /etc/hosts

Enter the following lines in the /etc/hosts file.

192.168.1.109 hadoop-master

192.168.1.145 hadoop-slave-1

192.168.56.1 hadoop-slave-2

Configuring Key Based Login

Set up ssh on each node so that they may interact without needing to enter a password.

$ suhadoop

$ ssh-keygen -t rsa

$ ssh-copy-id -i ~/.ssh/id_rsa.pub localhost@hadoop-master

$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_ap1t@hadoop-slave-1

$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_ap2t@hadoop-slave-2

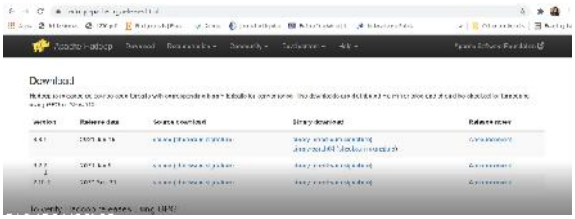
$ chmod 0600 ~/.ssh/authorized_keys

$ exit

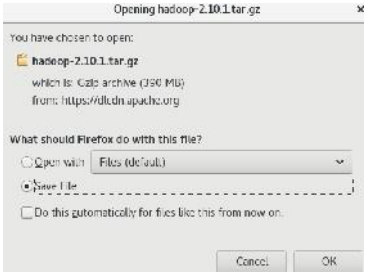
8.5 Installing Hadoop

Downloading and installing Hadoop

https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.10.1/hadoop-3.2.2.tar.gz



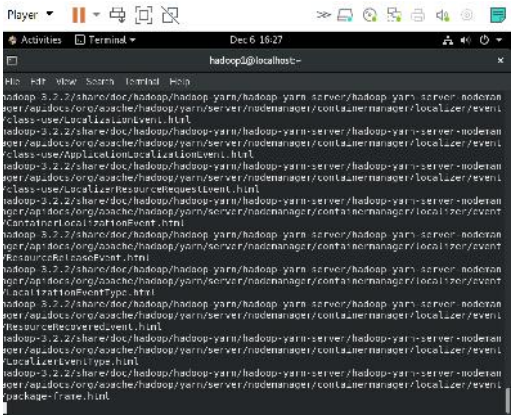
Make sure to download the binary file.



On the terminal, use the following command to extract the Hadoop file:

tar -xvf hadoop-3.2.2.tar.gz

Extracting hadoop file as shown in figure below:



Step12: Moving Hadoop to a Location

After using the ls command,we will notice Hadoop folder is created. In next step, we will rename the file using the mv command from Hadoop-3.2.2 to Hadoop.

Step13: Editing and Configuring HadoopYou must first set the path in the ~/.bashrc file. The command ~ /.bashrc can be used to set the path from the root user. You should check your Java configurations before editing ~/.bashrc.

update-alternatives-config java

You'll now be able to see all of the Java versions installed on the computer. Because I only have one version of Java, which is the most recent, it is displayed below:

You can also have several versions.

The next step is to choose the version you wish to work on. In the screenshot above, you can see a path that has been marked. Copy and paste this path into a gedit file. This route will only be utilised in the next phases.

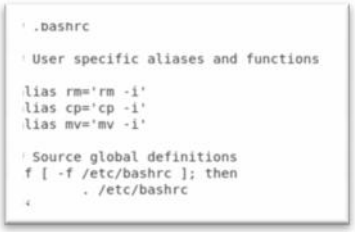
Enter the number of the selection you've made. I've picked number one in this case. Now use the vi editor to open /.bashrc (the screen-oriented text editor in Linux)

Note that you must first become a root user before editing /.bashrc.

When you get logged into your root user, enter the command:

vi ~/.bashrc

The command above should bring you to the vi editor, where you should see the following screen:



To get to this, hit the Insert key on your keyboard, and then start typing the following code to set a Java path:

fi

#HADOOP VARIABLES START

export JAVA_HOME= (path you copied in the previous step)

export HADOOP_HOME=/home/(your username)/hadoop

export PATH=$PATH:$HADOOP_HOME/bin

export PATH=$PATH:$HADOOP_HOME/sbin

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

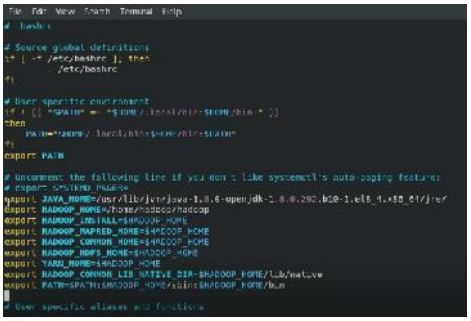
export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_/HOME/lib/native

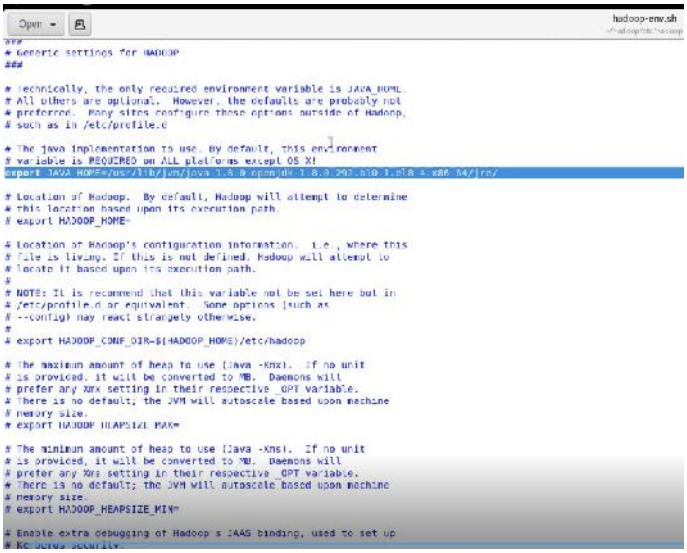
export HADOOP_OPTS="Djava.library.path"=$HADOOP_HOME/lib"

#HADOOP VARIABLES END

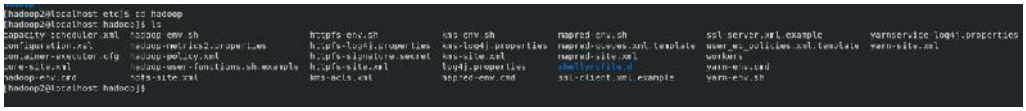
After writing the code, click on **Esc** on your keyboard and write the command: wq**!** This will save and exit you from the vi editor. The path has been set now as it can be seen in the image below:



Step14: Using the vi editor, open hadoop-env.sh. To inform Hadoop which path to use, replace this path with the Java path. You will be presented with the following window:



Step15: There are multiple XML files that need to be modified now, and you must specify the property and path for each one. All configuration files are shown in the image below:



- **Editing core-site.xml**
  - use the command below to open core-site.xml file

    vim core-site.xml
  - Enter the following code in between the configuration tags as below

  - Now, exit from this window by entering the command: wq!
- **Editing hdfs-site.xml**
  - Use the command below to open hdfs-site.xml

    vim hdfs-site.xml
  - Enter the following code in between the configuration tags as below

  - Exit using **Esc** and the command: wq**!**

- **Editing mapred-site.xml**
  - Use the command below to open hdfs-site.xml

    vim mapred-site.xml
  - Enter the following code in between the configuration tags as below:

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
        <name>mapreduce..job.tracker</name>
        <value>localhost:9001</value>
</property>
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
</property>
<property>
 <name>mapreduce.application.classpath</name>
 <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
</property>
</configuration>
```

  - E

    xit using **Esc** and the command: wq**!**

- **Editing yarn-site.xml**
  - Use the command below to open hdfs-site.xml

    vim yarn-site.xml

```
<configuration>


<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
        <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
  </property>
</configuration>
```

  - Exit from this window by pressing Esc and then writing the command: wq**!**

Step16: Create a directory namenode, datanode, and secondary using the command below:

```
[hadoop2@localhost var]$ sudo mkdir -p hadoop_warehouse/dfs/namenode
[sudo] password for hadoop2:
[hadoop2@localhost var]$ sudo mkdir -p hadoop_warehouse/dfs/datanode
[hadoop2@localhost var]$ sudo mkdir -p hadoop_warehouse/dfs/secondary
[hadoop2@localhost var]$ ls -ld hadoop_warehouse
drwxr-xr-x. 4 root root 28 Aug  4 15:35 hadoop_warehouse
[hadoop2@localhost var]$
```

Step 17: As we can see in the above image, permissions have been given only to the root. So, next, we will use chown command as shown in image below to change permission to hadoop2.

```
drwxr-xr-x. 4 root root 28 Aug  4 15:35 hadoop_warehouse
[hadoop2@localhost var]$ sudo chown hadoop2:hadoop2 -R hadoop_warehouse/
[hadoop2@localhost var]$ ls -ld hadoop_warehouse
drwxr-xr-x. 4 hadoop2 hadoop2 28 Aug  4 15:35 hadoop_warehouse
```

Step 18: To check permissions of all the file that comes under Hadoop_datawarehouse, following command will be executed:

```
[hadoop2@localhost var]$ cd hadoop_warehouse
[hadoop2@localhost hadoop_warehouse]$ ls
dfs  tmp
[hadoop2@localhost hadoop_warehouse]$ ll
total 0
drwxr-xr-x. 5 hadoop2 hadoop2 55 Aug  4 15:35 dfs
drwxr-xr-x. 3 hadoop2 hadoop2 20 Aug  4 15:26 tmp
[hadoop2@localhost hadoop_warehouse]$
```

All the files that comes into this folder their permission has changed.

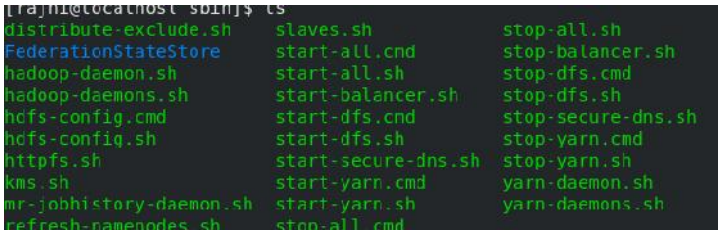
Step22: Lets go to the Hadoop directory and run the command as shown below to format the name node.

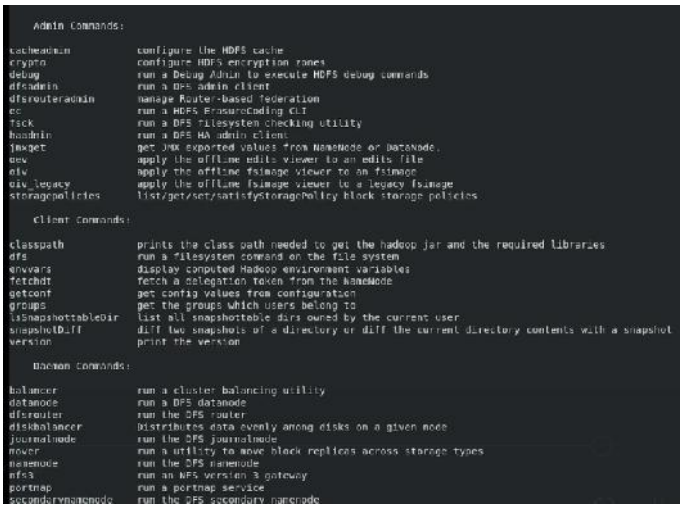
hadoopnamenode -format

So, we will get a message that namenode has been successfully formatted.

Step 19: To start all the services or start Hadoop daemons. To start services, we will go to sbin folder and will see all services.

start dfs.sh





Step20: Checking Hadoop

You must now verify whether the Hadoop installation was completed successfully on your machine.

Go to the directory where you extracted the Hadoop tar file, right-click on the bin, and select Open in Terminal from the menu.

Now type the command ls. If you get a window like the one below, that means Hadoop has been successfully installed!

```
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
  CLASSNAME            run the class named CLASSNAME
 or
  where COMMAND is one of:
  fs                   run a generic filesystem user client
  version              print the version
  jar <jar>            run a jar file
                       note: please use "yarn jar" to launch
                             YARN applications, not this command.
  checknative [-a|-h]  check native hadoop and compression libraries availabili
y
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop arch
ve
  classpath            prints the class path needed to get the
  credential           interact with credential providers
                       Hadoop jar and the required libraries
  daemonlog            get/set the log level for each daemon
  trace                view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
```

Summary

- Hadoop MapReduce is a programming paradigm used by Apache Hadoop to provide tremendous scalability across hundreds or thousands of Hadoop clusters running on cheap hardware. On a Hadoop cluster, the MapReduce paradigm uses a distributed algorithm to process huge unstructured data sets.
- The fundamental components of Apache Hadoop, which are incorporated into CDH and supported by a Cloudera Enterprise subscription, allow you to store and handle a limitless amount of data of any sort on a single platform.
- YARN is an open source resource management framework for Hadoop that allows you to go beyond batch processing and expose your data to a variety of workloads such as interactive SQL, sophisticated modelling, and real-time streaming.
- Hadoop's shuffle phase passes map output from a Mapper to a Reducer in MapReduce.
- In MapReduce, the sort phase is responsible for combining and sorting map outputs. The mapper's data is aggregated by key, distributed across reducers, then sorted by key. All values associated with the same key are obtained by each reducer.
- The JobTracker is a Hadoop service that distributes MapReduce tasks to specified nodes in the cluster, preferably those that hold the data or are in the same rack. Jobs are submitted to the Job Tracker by client apps. The JobTracker sends the work to the TaskTracker nodes that have been selected.
- Hadoop streaming is a feature included in the Hadoop distribution. You may use this programme to construct and run Map/Reduce tasks using any executable or script as the mapper and/or reducer.
- Ssh-keygen is an utility that allows you to generate fresh SSH authentication key pairs. This type of key pair is used to automate logins, provide single sign-on, and authenticate hosts.
- Hadoop is a Java-based Apache open source platform that allows big datasets to be processed across clusters of computers using simple programming techniques. The Hadoop framework application runs in a clustered computing environment that allows for distributed storage and computation.

Keywords

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

Job Tracker: The job tracker is a master daemon that runs on the same node as the data nodes and manages all of the jobs. This data will be stored on multiple data nodes, but it is the task tracker's responsibility to keep track of it.

Process is what daemons stand for. **Hadoop Daemons** are a collection of Hadoop processes. Because Hadoop is a Java platform, all of these processes are Java Processes.

Resource Manager: The Resource Manager in YARN is basically a scheduler. In essence, it is confined to dividing the system's available resources among competing applications. It optimises optimal cluster utilisation (keeping all resources in use at all times) while taking into account different limitations such as capacity guarantees, fairness, and service level agreements (SLAs). The Resource Manager contains a pluggable scheduler that permits other algorithms, such as capacity, to be utilised as needed to accommodate varied policy restrictions. The "yarn" user is used by the daemon.

Application Master: The Application Master is a framework-specific library that is in charge of negotiating resources with the Resource Manager and working with the Node Manager(s) to execute and monitor Containers and their resource usage. It is in charge of negotiating suitable resource Containers with the Resource Manager and keeping track of their progress. The Resource Manager monitors the Application Master, which operates as a single Container.

Job history service: This is a daemon that keeps track of jobs that have been finished. It's best to run it as a separate daemon. Because it maintains task history information, running this daemon uses a lot of HDFS space. The "mapred" user runs this daemon.

Container: A Container is a resource allocation that occurs as a result of a Resource Request being granted by the Resource Manager. A Container allows a programme to access a certain amount of resources (memory, CPU, etc.) on a certain host. To make use of Container resources, the Application Master must take the Container and offer it to the Node Manager in charge of the host to which the Container has been assigned. To guarantee that Application Master(s) cannot fake allocations in the cluster, the Container allocation is checked in secure mode.

Node Manager:A Container is a resource allocation that occurs as a result of a Resource Request being granted by the Resource Manager. A Container allows a programme to access a certain amount of resources (memory, CPU, etc.) on a certain host. To make use of Container resources, the Application Master must take the Container and offer it to the Node Manager in charge of the host to which the Container has been assigned. To guarantee that Application Master(s) cannot fake allocations in the cluster, the Container allocation is checked in secure mode.

NameNode is a component of the Master System. Namenode's main function is to manage all of the MetaData. The list of files saved in HDFS is known as metadata (Hadoop Distributed File System). In a Hadoop cluster, data is stored in the form of blocks, as we all know.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Data node: The data node is a commodity computer with the GNU/Linux operating system and data node software installed. In a cluster, there will be a data node for each node (common hardware/system).

Map-red: It is one of the most significant configuration files for Hadoop's runtime environment settings. It includes MapReduce's setup options. By setting the MapReduce.framework.name variable in this file, we may give MapReduce a name.

Self Assessment

1. Which of the following are major pre-requisites for MapReduce programming.

A. The application must lend itself to parallel programming

B. The data for the applications can be expressed in key-value pairs

C. Both

D. None of above

2. _____ maps input key/value pairs to a set of intermediate key/value pairs.

A. Mapper

B. Reducer

C. Both Mapper and Reducer

D. None of the mentioned

3. Input key/value pairs are mapped to a collection of intermediate key/value pairs using _____ .

A. Mapper

B. Reducer

C. Both Mapper and Reducer

D. None of the mentioned

4. The master is a _____, and each cluster has only one NameNode.

A. Data Node

B. NameNode

C. Data block

D. Replication

5. HDFS is written in the _____ programming language.

A. C++

B. Java

C. Scala

D. None of the mentioned

6. The reduce step will read the _____ results.

A. Intermediate

B. Complete

C. Initial

D. None of above

7. Map step reads data in _____ pair format

A. Input-output

B. Key-Value

C. Read-write

D. None of above

8. _____ is the main prerequisite for Hadoop.

A. C

B. C++

C. C#

D. Java

9. Commands to create a system user account on both master and slave systems

A. useraddhadoop

B. adduserhadoop

C. useridhadoop

D. addidHadoop

10. Hadoop Streaming uses standard ____ streams as the interface between Hadoop and user program.

A. Unix

B. Linux

C. C++

D. None of above

11. Please identify the right statement.

A. You can specify any executable as the mapper and/or the reducer

B. You cannot supply a Java class as the mapper and/or the reducer.

C. The class you supply for the output format should return key/value pairs of Text class

D. All of the mentioned

12. Which of the Hadoop streaming command option arguments is mandatory?

A. output directoryname

B. mapper executable

C. input directory name

D. all of the mentioned

13. You have to edit hosts file in _____ folder

A. /etc/

B. /java/

C. /Hadoop/

D. /base/

14. The Aggregate package provides which of the following classes?

A. Class1

B. Reducer

C. Reducer2

D. Reducer3

15. JPS command is used to check

A. if a specific daemon is up or not.

B. if java is installed or not

C. for one specific daemon.

D. None of above.

Answers for Self Assessment

1.	C	2.	A	3.	A	4.	B	5.	B
6.	A	7.	B	8.	D	9.	A	10.	A
11.	A	12.	D	13.	A	14.	B	15.	A

Review Questions

1. Difference between job tracker and task tracker.
2. Write down steps to install hadoop.
3. Write down HDFS components.
4. What do you understand by resource manager.
5. What is function of. /bashrc file?

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practice of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley. White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

Unit 09: Hadoop Node Commands

CONTENTS

Objectives

- Starting HDFS
- Creating User Account
- Configuring Key Based Login
- Configuring Hadoop on Master Server
- Understand how to add a New Data Node in the Hadoop Cluster
- Learn Adding User and Ssh Access.

Introduction

Starting HDFS

Format the configured HDFS file system and then open the namenode (HDFS server) and execute the following command.

hadoopnamenode -format

Namenode is the node in the Hadoop Distributed File System which keeps track of all the data stored in the Datanode. Namenode has metadata related to the data stored on the Datanodes and has information related to the location of the data stored. So, when you run the **hadoopnamenode -format** command, all these information is deleted from the namenode which means that the system does not know where the data is stored hence losing all data stored in the Hadoop File System.Formatting the namenode deletes the information from namenode directory. The NameNode directory is specified in hdfs-site.xml file in

dfs.namenode.name.dir property.

Formatting the file system means initializing the directory specified by the dfs.name.dir variable.After you have logged in as the dedicated user for Hadoop(in my case it is hduser) that you must have created while installation, go to the installation folder of Hadoop(in my case it is /usr/local/hadoop).

start-dfs.sh

Inside the directory Hadoop, there will be a folder 'sbin', where there will be several files like start-all.sh, stop-all.sh, start-dfs.sh, stop-dfs.sh, hadoop-daemons.sh, yarn-daemons.sh, etc. Executing these files can help us start and/or stop in various ways. start-all.sh & stop-all.sh: Used to start and stop hadoop daemons all at once. Issuing it on the master machine will start/stop the daemons on all the nodes of a cluster. These commands are now deprecated. start-dfs.sh, stop-dfs.sh and start-yarn.sh, stop-yarn.sh: Same as above but start/stop HDFS and YARN daemons separately on all the nodes from the master machine. It is advisable to use these commands instead of start-all.sh & stop-all.sh. To start individual daemons on an individual machine manually. You need to go to a particular node and supply these commands. hadoop-daemon.sh namenode/datanode and yarn-deamon.sh resourcemanager.

HDFS Operations to Read & Write the Data

You can execute almost all operations on Hadoop Distributed File Systems that can be executed on the local file system. You can execute various reading, writing operations such as create a directory, provide permissions, copy files, update files, delete, etc. You can add access rights and browse the file system to get the cluster information like the number of dead nodes, live nodes, spaces used, etc.

- ***HDFS Operations to Read the file***

  To read any file from the HDFS, you have to interact with the NameNode as it stores the metadata about the DataNodes. The user gets a token from the NameNode and that specifies the address where the data is stored.You can put a read request to NameNode for a particular block location through distributed file systems. The NameNode will then check your privilege to access the DataNode and allows you to read the address block if the access is valid.

$ hadoop fs -cat <file>

The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS) as well as other file systems that Hadoop supports, such as Local FS, WebHDFS, S3 FS, and others

HDFS Operations to write in file

Similar to the read operation, the HDFS Write operation is used to write the file on a particular address through the NameNode. This NameNode provides the slave address where the client/user can write or add data. After writing on the block location, the slave replicates that block and copies to another slave location using the factor 3 replication. The salve is then reverted back to the client for authentication.

bin/hdfs dfs -ls <path>

The process for accessing a NameNode is pretty similar to that of a reading operation. Below is the HDFS write commence:

- **ls:** This command is used to list all the files. Use *lsr* for recursive approach. It is useful when we want a hierarchy of a folder.It will print all the directories present in HDFS. bin directory contains executables so, *bin/hdfs* means we want the executables of hdfs particularly *dfs*(Distributed File System) commands.

Listing Files in HDFS

Finding the list of files in a directory and the status of a file using 'ls' command in the terminal. Syntax of ls can be passed to a directory or a filename as an argument which are displayed as follows:

$HADOOP_HOME/bin/hadoop fs -ls <args>

All hadoop commands are invoked by the bin/hadoop script. Running the hadoop script without any arguments prints the description for all commands

Inserting Data into HDFS

Step 1. Create an input directory

$HADOOP_HOME/bin/hadoop fs -mkdir /user/input

Step 2. Use put command transfer and store the data file from the local systems to the HDFS using the following commands in the terminal

$HADOOP_HOME/bin/hadoop fs -put /home/abc.txt /user/input

Step 3. Verify the file using ls command.

$HADOOP_HOME/bin/hadoop fs -ls /user/input

Retrieving Data from HDFS

For an instance if you have a file in HDFS called abc. Then retrieve the required file from the Hadoop file system by carrying out:

Step 1. View the data from HDFS using cat command.

$HADOOP_HOME/bin/hadoop fs -cat /user/output/abc

Step 2. Gets the file from HDFS to the local file system using get command as shown below

$HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/

Shutting down the HDFS

Step 3. Shut down the HDFS files by following the below command

$ stop-dfs.sh

Here dfs daemons are stopped separately

Multi-Node Cluster

A Multi Node Cluster in Hadoop contains two or more DataNodes in a distributed Hadoop environment. This is practically used in organizations to store and analyze their Petabytes and Exabytes of data. Here, we are taking two machines – master and slave. On both the machines, a Datanode will be running.Installing Java

Syntax of java version command

$ java –version

9.1 Creating User Account

System user account is used on both master and slave systems for the Hadoop installation.

useraddhadoop

passwd hadoop

Mapping the nodes

- Hosts files should be edited in /etc/ folder on each and every node and IP address of each system followed by their host names must be specified mandatorily.

vi /etc/hosts

Map-Only job in the Hadoop is the process in which **mapper** does all tasks. No task is done by the **reducer**. Mapper's output is the final output. MapReduce is the data processing layer of Hadoop. It processes large structured and unstructured data stored in **HDFS**. MapReduce also processes a huge amount of data in parallel. It does this by dividing the job (submitted job) into a set of independent tasks (sub-job). In Hadoop, MapReduce works by breaking the processing into phases: **Map** and **Reduce**.

- **Map:** It is the first phase of processing, where we specify all the complex logic code. It takes a set of data and converts into another set of data. It breaks each individual element into tuples (**key-value pairs**).
- **Reduce:** It is the second phase of processing. Here we specify light-weight processing like aggregation/summation. It takes the output from the map as input. Then it combines those tuples based on the key. Using a text editor, open the hosts file on every host in your cluster. A Hosts file is a file that almost all computers and operating systems **can use to map a connection between an IP address and domain names**. This file is an ASCII text file. It contains IP addresses separated by a space and then a domain name.

Enter the following lines in the /etc/hosts file.

192.168.1.109 hadoop-master

192.168.1.145 hadoop-slave-1

192.168.56.1 hadoop-slave-2

9.2 Configuring Key Based Login

Ssh should be setup in each node so they can easily converse with one another without any prompt for password.

suhadoop

$ ssh-keygen -t rsa

$ ssh-copy-id -i ~/.ssh/id_rsa.pub abc@hadoop-master

$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_tp1@hadoop-slave-1

$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_tp2@hadoop-slave-2

$ chmod 0600 ~/.ssh/authorized_keys

$ exit

So, you have spent your time in pseudo mode and you have finally started moving to your own cluster? Perhaps you just jumped right into the cluster setup? In any case, a distributed Hadoop cluster setup requires your "master" node [name node & job tracker] to be able to SSH (without requiring a password, so key based authentication) to all other "slave" nodes (e.g. data nodes).The need for SSH Key based authentication is required so that the master node can then login to slave nodes (and the secondary node) to start/stop them, etc. This is also required to be setup on the secondary name node (which is listed in your masters file) so that [presuming it is running on another machine which is a ***very*** goodidea for a production cluster] will be started from your name node with

./start-dfs.sh

and job tracker node with

./start-mapred.sh

Now you need to copy (however you want to-do this please go ahead) your public key to all of your "slave" machine (don't forget your secondary name node). It is possible (depending on if these are new machines) that the slave's **hadoop** user does not have a .ssh directory and if not you should create it (**$ mkdir ~/.ssh**). You need to verify the permissions of the authorized_keys file and the folder / parent folders in which it is located.

9.3 Installation of Hadoop

Hadoop should be downloaded in the master server using the following procedure. To create a directory

mkdir /opt/Hadoop

cd /opt/hadoop/

The wget command is a **command line utility for downloading files from the Internet**. It supports downloading multiple files, downloading in the background, resuming downloads, limiting the bandwidth used for downloads and viewing headers

wget http://apache.mesi.com.ar/hadoop/common/hadoop-1.2.1/hadoop-1.2.0.tar.gz

The tar command is **used to create compressed archives which represent a file or collection of files**. A tar file, commonly known as a "tarball," a gzip, or a bzip file, will have an extension ending with . tar or . tar. gz

tar -xzf hadoop-1.2.0.tar.gz

mv stands for **move**. mv is used to move one or more files or directories from one place to another in a file system

mv hadoop-1.2.0 hadoop

chown -R hadoop /opt/hadoop

cd /opt/hadoop/hadoop/

It has two distinct functions:
(i) It renames a file or folder.
(ii) It moves a group of files to a different directory.
No additional space is consumed on a disk during renaming. This command normally works silently means no prompt for confirmation.

The chown command allows you to change the user and/or group ownership of a given file. **chown** command is used to change the file Owner or group. Whenever you want to change ownership you can use chown command.

chown -R hadoop /opt/hadoop

cd /opt/hadoop/hadoop/

-R, --recursive: It is used to perform operations on files and directories recursively.

cd command in linux known as **change directory command**. It is used to change current working directory.

cd /opt/hadoop/hadoop/

Configuring Hadoop

Hadoop server must be configured in core-site.xml and should be edited where ever required.

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://hadoop-master:9000/</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

The core-site.xml file informs Hadoop daemon where NameNode runs in the cluster. It contains the configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.

- The hdfs-site.xml file contains the configuration settings for HDFS daemons; the NameNode, the Secondary NameNode, and the DataNodes. Here, we can configure hdfs-

site.xml to specify default block replication and permission checking on HDFS. The actual number of replications can also be specified when the file is created. The default is used if

<configuration> <property> <name>dfs.data.dir</name> <value>/opt/hadoop/hadoop/dfs/name/data</value> <final>true</final> </property>	<property> <name>dfs.name.dir</name> <value>/opt/hadoop/hadoop/dfs/name</value> <final>true</final> </property> <property> <name>dfs.replication</name> <value>1</value> </property> </configuration>

replication is not specified in create time.Hadoop server must be configured in hdfs-site.xml and should be edited where ever required.

Hadoop server must be configured in mapred-site.xml and should be edited where ever required.

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>hadoop-master:9001</value>
</property>
</configuration>
```

The mapred-site.xml file contains the configuration settings for MapReduce daemons; the job tracker and the task-trackers.

hadoop-env.sh

JAVA_HOME, HADOOP_CONF_DIR, and HADOOP_OPTS should be edited as follows:

```
export JAVA_HOME=/opt/jdk1.7.0_17
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_CONF_DIR=/opt/hadoop/hadoop/conf
```

Installing Hadoop on slave servers

HADOOP_CONF_DIR is the environment variable that set the directory location.

```
export JAVA_HOME=/opt/jdk1.7.0_17(The java implementation to use.  Overrides JAVA_HOME)
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_CONF_DIR=/opt/hadoop/hadoop/conf
```

Extra Java runtime options.

```
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_CONF_DIR=/opt/hadoop/hadoop/conf
```

Installing Hadoop on Slave Servers

- DataNodes are the slave nodes in HDFS. Unlike NameNode, DataNode is a commodity hardware, that is, a non-expensive system which is not of high quality or high-availability. The DataNode is a block server that stores the data in the local file ext3 or ext4.These are slave daemons or process which runs on each slave machine.
- The actual data is stored on DataNodes. The DataNodes perform the low-level read and write requests from the file system's clients. They send heartbeats to the NameNode periodically to report the overall health of HDFS, by default, this frequency is set to 3 seconds.Hadoop should be installed on all the slave servers

```
# suhadoop
$ cd /opt/hadoop
$ scp -r hadoop hadoop-slave-1:/opt/hadoop
$ scp -r hadoop hadoop-slave-2:/opt/hadoop
```

SCP (secure copy) is a **command**-line utility that allows you to securely copy files and directories between two locations. scp can only access the file system of individual nodes within the cluster. It can't be used toaccess data in the HDFS-compatible storage for the cluster. Use scp when you need to upload a resource for use from an SSH session. For example, upload a Python script and then run the script from an SSH session.

Configuring Hadoop on Master Server

Key Role of masters is to store large amount of data on HDFS & perform parallel computation on this data using Map Reduce programs (Algorithms).NameNode, Secondary NameNode & Job Tracker are the masters servers in Hadoop.Master Servers are CPU/Memory intensive.Master server configuration

```
# suhadoop
$ cd /opt/hadoop/hadoop
```

Master Node Configuration

```
$ vi etc/hadoop/masters
hadoop-master
```

Slave Node Configuration

```
$ vi etc/hadoop/slaves
hadoop-slave-1
hadoop-slave-2
```

Name Node format on Hadoop Master

```
# su hadoop
$ cd /opt/hadoop/hadoop
$ bin/hadoop namenode –format
```

Master node in a hadoop cluster is responsible for storing data in HDFS and executing parallel computation the stored data using MapReduce. Master Node has 3 nodes – NameNode, Secondary NameNode and JobTracker. JobTracker monitors the parallel processing of data using MapReduce while the NameNode handles the data storage function with HDFS. NameNode keeps a track of all the information on files (i.e. the metadata on files) such as the access time of the file, which user is accessing a file on current time and which file is saved in which hadoop cluster. The secondary NameNode keeps a backup of the NameNode data.

Slave/Worker Node- This component in a hadoop cluster is responsible for storing the data and performing computations. Every slave/worker node runs both a TaskTracker and a DataNode service to communicate with the Master node in the cluster. The DataNode service is secondary to the NameNode and the TaskTracker service is secondary to the JobTracker.

9.4 Hadoop Services

Starting Hadoop services on the Hadoop-Master procedure explains its setup.

$ cd $HADOOP_HOME/sbin

$ start-all.sh

This will startup a Name node, Data node, Job tracker and a Task tracker on your machine.

start-all.sh & stop-all.sh: Used to start and stop hadoop daemons all at once. Issuing it on the master machine will start/stop the daemons on all the nodes of a cluster. Deprecated as you have already noticed.

start-dfs.sh, stop-dfs.sh and start-yarn.sh, stop-yarn.sh : Same as above but start/stop HDFS and YARN daemons separately on all the nodes from the master machine. It is advisable to use these commands now over start-all.sh & stop-all.sh

hadoop-daemon.sh namenode/datanode and yarn-deamon.sh resourcemanager : To start individual daemons on an individual machine manually. You need to go to a particular node and issue these commands.

In a multi-node hadoop cluster, all the essential daemons are up and run on different machines/hosts. A multi-node hadoop cluster setup has a master slave architecture where in one machine acts as a master that runs the NameNode daemon while the other machines acts as slave or worker nodes to run other hadoop daemons. Usually in a multi-node hadoop cluster there are cheaper machines (commodity computers) that run the TaskTracker and DataNode daemons while other services are run on powerful servers. For a multi-node hadoop cluster, machines or computers can be present in any location irrespective of the location of the physical server.

9.5 Adding a New Data Node in the Hadoop Cluster

Given below are the steps to be followed for adding new nodes to a Hadoop cluster.

Networking

Add new nodes to an existing Hadoop cluster with some appropriate network configuration. Assume the following network configuration.

For New node Configuration –

IP address: 192.168.1.101

Netmask: 255.255.255.0

hostname: slave123.in

Adding User and SSH Access

- Add a User

  On a new node, add "hadoop" user and set password of Hadoop user to "hadoop123" or anything you want.

  useraddhadoop

  passwd hadoop

- Setup Password less connectivity from master to new slave.
- Execute the following on the master

  mkdir -p $HOME/.ssh

  chmod 700 $HOME/.ssh

  ssh-keygen -t rsa -P '' -f $HOME/.ssh/id_rsa

  cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

  chmod 644 $HOME/.ssh/authorized_keys

  Copy the public key to new slave node in hadoop user $HOME directory

scp $HOME/ .ssh/id_rsa.pub hadoop@192.168.1.101

- **Execute the following on the slaves**

  Login to hadoop. If not, login to hadoop user.

  suhadoopssh -X hadoop@192.168.1.101

- Copy the content of public key into file **"$HOME/.ssh/authorized_keys"** and then change the permission for the same by executing the commands.

```
cd $HOME
mkdir -p $HOME/ .ssh
chmod 700 $HOME/ .ssh
cat id_rsa.pub >>$HOME/.ssh/authorized_keys
chmod 644 $HOME/ .ssh/authorized_keys
```

- Check ssh login from the master machine. Now check if you can ssh to the new node without a password from the master.

  sshhadoop@192.168.1.101 or hadoop@slave123

Set Hostname of New Node

You can set hostname in file **/etc/sysconfig/network.**

On new slave123 machine

```
Networking=yes
Hostaname=slave123.in
```

To make the changes effective, either restart the machine or run hostname command to a new machine with the respective hostname (restart is a good option).

On slave3 node machine −

```
hostname slave3.in
```

Update **/etc/hosts** on all machines of the cluster with the following lines –

```
192.168.1.100 slave2.in slave2
```

- Now try to ping the machine with hostnames to check whether it is resolving to IP or not.
- On new node machine –

ping master.in

Start the DataNode on New Node

Start the datanode daemon manually using **$HADOOP_HOME/bin/hadoop-daemon.sh script**. It will automatically contact the master (NameNode) and join the cluster. We should also add the new node to the conf/slaves file in the master server. The script-based commands will recognize the new node.

Login to new node

```
suhadoop or ssh -X hadoop@192.168.1.101
```

Start HDFS on a newly added slave node by using the following command

```
./bin/hadoop-daemon.sh start datanode
```

Check the output of jps command on a new node. It looks as follows.

```
$jps
7141 Datanode
10312 Jps
```

Removing a DataNode from the Hadoop Cluster

We can remove a node from a cluster on the fly, while it is running, without any data loss. HDFS provides a decommissioning feature, which ensures that removing a node is performed safely. To use it, follow the steps as given below –

Step 1 − Login to master

Login to master machine user where Hadoop is installed.

```
$suhadoop
```

Step2 – Change cluster configuration

An exclude file must be configured before starting the cluster. Add a key named dfs.hosts.exclude to our **$HADOOP_HOME/etc/hadoop/hdfs-site.xml** file. The value associated with this key provides the full path to a file on the NameNode's local file system which contains a list of machines which are not permitted to connect to HDFS.

For example, add these lines to **etc/hadoop/hdfs-site.xml** file.

```
<property>
        <name>dfs.hosts.exclude</name>
        <value>/home/hadoop/hadoop-1.2.1/hdfs_exclude.txt</value>
        <description>DFS exclude</description>
</property>
```

Step3- Determine hosts to decommission

One domain name per line should be added to the hdfs exclude.txt file for each computer to be retired. They will be unable to connect to the NameNode as a result of this. If you wish to delete DataNode2, look at the contents of the "/home/hadoop/hadoop-1.2.1/hdfs exclude.txt" file.

Slave2.in

Step4: Force configuration reload

Run the command "$HADOOP_HOME/bin/hadoopdfsadmin -refreshNodes" without the quotes.

```
$HADOOP_HOME/bin/hadoopdfsadmin -refreshNodes
```

The NameNode will be forced to re-read its configuration, including the newly modified 'excludes' file. It will decommission the nodes over time, giving enough time for each node's blocks to be duplicated onto active machines.

Examine the output of the jps command on slave2.in. After some time has passed, you will notice that the DataNode process has been automatically terminated.

Step5: Shutdown Nodes

The retired hardware can be securely turned down for maintenance when the decommissioning process is done. To verify the status of decommissioning, use the report command to dfsadmin. The state of the decommission node and the cluster's associated nodes will be described by the following command.

```
$HADOOP_HOME/bin/hadoopdfsadmin -report
```

Step6: Edit excludes file again

The machines can be deleted from the 'excludes' file after they have been decommissioned. Running "$HADOOP HOME/bin/hadoopdfsadmin -refreshNodes" again will read the exclusions file back into the NameNode, allowing the DataNodes to rejoin the cluster when the maintenance is complete, or if extra capacity is required in the cluster, etc.

Special Note: If the aforementioned procedure is done but the tasktracker process remains active on the node, it must be terminated. Disconnecting the machine, like we did in the previous phases, is one option. The Master will immediately recognise the process and proclaim it dead. Because the tasktracker isn't as important as the DataNode, there's no need to go through the same steps to

remove it. DataNode holds the data that you wish to delete safely and without losing any information.

The following command may be used to start/stop the tasktracker on the fly at any moment.

$HADOOP_HOME/bin/hadoop-daemon.sh stop tasktracker

$HADOOP_HOME/bin/hadoop-daemon.sh strattasktracker

Summary

- Hadoop applications use the Hadoop Distributed File Solution (HDFS) as their primary data storage system. HDFS is a distributed file system that uses a NameNode and DataNode architecture to allow high-performance data access across highly scalable Hadoop clusters.
- Hadoop is an Apache Foundation open source framework for processing huge volumes of heterogeneous data sets in a distributed way across clusters of commodity computers and hardware using a simplified programming style. Hadoop is a dependable distributed storage and analysis solution.
- By duplicating data over numerous nodes, it provides extremely stable and distributed storage that assures reliability even on commodity hardware. When data is submitted to HDFS, it is automatically divided into numerous blocks (adjustable parameter) and stored/replicated across many data nodes, unlike a traditional file system. As a result, high availability and fault tolerance are ensured.
- MapReduce is a data analysis system that can handle enormous datasets and conduct sophisticated computations. This component is in charge of all calculations, and it does so by breaking down a big complicated computation into numerous tasks and assigning them to individual worker/slave nodes, as well as coordinating and consolidating the results.
- Namenode stores information about all other nodes in the Hadoop Cluster, files in the cluster, file component blocks and their positions in the cluster, and other information that is necessary for the Hadoop Cluster's functioning.
- Job Tracker manages the sharing of information and outcomes by keeping track of the specific tasks/jobs allocated to each of the nodes.
- Tracker is in charge of completing the job or computation that has been assigned to it.
- Datanode is in charge of storing the information.

Keywords

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

Job Tracker: The job tracker is a master daemon that runs on the same node as the data nodes and manages all of the jobs. This data will be stored on multiple data nodes, but it is the task tracker's responsibility to keep track of it.

Failover: If the primary system fails or is taken down for maintenance, failover is a backup operational mode that immediately switches to a standby database, server, or network. Failover technology smoothly sends requests from a downed or failing system to a backup system that replicates the operating system environment.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Name node: The name node is a piece of commodity hardware that houses the GNU/Linux operating system as well as name node software. It's a piece of software that can run on standard hardware.

Resource Manager: The Resource Manager in YARN is basically a scheduler. In essence, it is confined to dividing the system's available resources among competing applications. It optimises optimal cluster utilisation (keeping all resources in use at all times) while taking into account different limitations such as capacity guarantees, fairness, and service level agreements (SLAs). The Resource Manager contains a pluggable scheduler that permits other algorithms, such as capacity, to be utilised as needed to accommodate varied policy restrictions. The "yarn" user is used by the daemon.

Application Master: The Application Master is a framework-specific library that is in charge of negotiating resources with the Resource Manager and working with the Node Manager(s) to execute and monitor Containers and their resource usage. It is in charge of negotiating suitable resource Containers with the Resource Manager and keeping track of their progress. The Resource Manager monitors the Application Master, which operates as a single Container.

HBase: HBase is a Hadoop-based open-source database with sorted map data. It's horizontally scalable and column-oriented.

Blocks: Large files were broken into little segments known as Blocks in Hadoop HDFS. The physical representation of data is called a block. Except for the final block, which might be the same size or less, all HDFS blocks are the same size. Hadoop divides files into 128 MB blocks before storing them in the Hadoop file system.

Key/Value Store:A key-value store, sometimes known as a key-value database, is a simple database that employs an associative array (think of a map or dictionary) as its basic data model, with each key corresponding to one and only one item in a collection. A key-value pair is the name for this type of connection.

Review Questions

1. Select the command to format the configured HDFS file system

A. hadoopnamenode -format

B. hadoop -format namenode

C. hadoop name -format

D. hadoop node -format

2. Select the command to starts the hadoopdfs daemons, the namenode and datanodes.

A. start-mapred.sh

B. start-dfs.sh

C. hadoop-env.sh

D. start-daemons.sh

3. Select the command to stops the hadoopdfs daemons

A. stop-mapred.sh

B. stop-dfs.sh

C. stop-env.sh

D. stop-daemons.sh

4. Select the command to starts the hadoop map/reduce daemons, the jobtracker and tasktrackers.

A. start-mapred.sh

B. start-dfs.sh

C. start-env.sh
D. start-daemons.sh

5. Select the command to starts all Hadoop daemons
A. start-all-mapred.sh
B. start-all.sh
C. start-all-dfs.sh
D. start-daemons.sh

6. Which of the following are main layers of HDFS.
A. namespace layer and block storage service layer
B. datanode layer and block storage service layer
C. namenodelayer and block storage service layer
D. datanode layer and namenodelayer

7. The data nodes are used as _____ for blocks by all the namenodes
A. Common points
B. common storage
C. Both
D. None of above

8. A _____ is a set of blocks that belong to a single namespace.
A. Data Pool
B. Name Pool
C. Block Pool
D. All of above

9. A _____ identifier is used to identify all the nodes in the cluster.
A. NameID
B. ClusterID
C. DataID
D. None of above

10. _____ method clears all keys from the configuration.
A. Clear
B. addResource
C. getClass
D. none of the mentioned

11. Select the command to format the configured HDFS file system
A. hadoopnamenode -format
B. hadoop -format namenode
C. hadoop name -format

D. hadoop node -format

12. Select the command to starts the hadoopdfs daemons, the namenode and datanodes.

A. start-mapred.sh

B. start-dfs.sh

C. hadoop-env.sh

D. start-daemons.sh

13. Select the command to stops the hadoopdfs daemons

A. stop-mapred.sh

B. stop-dfs.sh

C. stop-env.sh

D. stop-daemons.sh

14. Select the command to starts the hadoop map/reduce daemons, the jobtracker and tasktrackers.

A. start-mapred.sh

B. start-dfs.sh

C. start-env.sh

D. start-daemons.sh

15. Select the command to starts all Hadoop daemons

A. start-all-mapred.sh

B. start-all.sh

C. start-all-dfs.sh

D. start-daemons.sh

16. Write down commands and explanation to insert and retrieve data into HDFS.

17. Explain HDFS operation to read and write the data.

18. Write down steps for learning adding user and ssh access.

19. Write down command to explain how to create user account.

20. Explain HDFS commands.

Answers for Self Assessment

1.	A	2.	B	3.	B	4.	A	5.	B
6.	A	7.	B	8.	C	9.	B	10.	A
11.	A	12.	B	13.	B	14.	A	15.	B

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think. Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier for Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
- of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/
7. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm

Unit 10: Map Reduce Applications

CONTENTS

Objectives

- Learn what is unit testing
- Explore concepts of MRUnit
- Learn Developing and testing MapReduce jobs with MRUnit
- Learn Anatomy of a MapReduce Job Run
- explore and learn the concepts of Shuffle and Sort

Introduction

Unit testing is a testing technique in which individual modules are checked by the developer to see if there are any flaws. It is concerned with the standalone modules' functional soundness.The basic goal is to isolate each component of the system in order to detect, analyse, and correct any flaws.

Unit Testing - Advantages:

Reduces bugs when modifying existing functionality or reducing defects in newly developed features. Defects are caught early in the testing process, which lowers the cost of testing. Enhances code reworking and improves design. When unit tests are used in conjunction with the build process, the build's quality is improved.

Unit Testing Techniques:

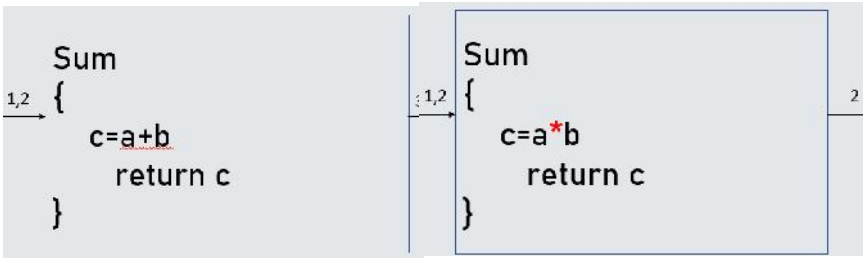
- **Black Box Testing -** The user interface, input, and output are all tested with this tool.
- **White Box Testing -** Each of the functions' behaviour is checked using this method.
- **Gray Box Testing -** Tests, hazards, and assessment procedures are all carried out with this tool.

If you are a java developer, then you have already worked with JUNIT framework

```
Sum
{
c=a+b
return c
}
```

JUNIT

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit that originated with JUnit. JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, taest a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.



10.1 What is MRUNIT

MRUnit is a Java library based on JUnit that lets us unit test Hadoop MapReduce programmes. This makes Hadoop MapReduce code bases simple to write and maintain. Mappers and Reducers can be tested separately in MRUnit, as well as MapReduce computations as a whole.Cloudera has created a testing library for MapReduce. MapReduce and standard testing tools can be easily integrated (e.g. JUnit). Apache MRUnit is an Open Source package that enables Hadoop Mappers, Reducers, and MapReduce programmes to be unit-tested. It facilitates the integration of MapReduce with standard testing libraries like as JUnit and Mockito, and aids in the bridging of the gap between MapReduce programmes and those traditional libraries (by providing a set of interfaces and test harnesses). It complements rather than replaces JUnit.Please be informed that knowledge of Hadoop MapReduce and JUnit is essential for a better understanding of this topic before continuing.

The following are MRUnit's three core classes:

- **MapDriver**: The MapDriver class is in charge of calling the Mapper's map() method.
- **ReducerDriver:** This is the driver class that calls the Reducer's reduce() method.
- **MapReduceDriver**: The combined MapReduce driver is in charge of first invoking the Mapper's map() function, then performing an in-memory Shuffle phase. The Reducer's reduce() function is called at the end of this phase.

Each of the classes listed above provides methods for providing test inputs and anticipated outcomes. The setup() method of the JUnit API is responsible for establishing fresh instances of the Mapper, Reducer, and relevant MRUnit drivers for each test.To include MRUnit in a Hadoop MapReduce project, include it as a test dependent in the project POM file (of course, the project is a Maven project, and I'm sure you're not going to skip Maven for this type of project):

```
<dependency>
<groupId>org.apache.mrunit</groupId>
<artifactId>mrunit</artifactId>
<version>1.1.0</version>
<classifier>hadoop2</classifier>
```

<scope>test</scope>

</dependency>

JUnit, of course, should also be present:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

In this debate, I am not referring to any specific IDE. A shell may create and manage the sample project using Maven. You may then select to import it into your preferred application.

The popular word counter is the MapReduce application under test (the Hello World of MapReduce). It reads text files and counts how many times each word appears. You may find hundreds of connections to this example on the internet, but just in case, here is the code for the Mapper.

```
public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
   public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
     String w = value.toString();
context.write(new Text(w), new IntWritable(1));
   }
}
```

and the Reducer

```
public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
  public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritableval : values) {
      sum += val.get();
    }
context.write(key, new IntWritable(sum));
  }
}
```

Let's get started with an MRUnit test case. It follows the same format as a JUnit test case. First, specify the instance variable that will be used in the MapReduce test. They contain the following MRUnit framework drivers:

```
private Mapper mapper;
private Reducer reducer;
private MapDrivermapDriver;
private ReduceDriverreduceDriver;
private MapReduceDrivermapReduceDriver;
```

Then create instances of them in the JUnit setUp() method:

```
@Before
public void setUp() throws Exception {
    mapper = new WordCountMapper();
    reducer = new WordCountReducer();
mapDriver = new MapDriver(mapper);
reduceDriver = new ReduceDriver(reducer);
mapReduceDriver = new MapReduceDriver(mapper, reducer);
}
```

To test the Mapper, just pass the inputs and anticipated outputs to the MapperDriver instance, then use the runTest() method:

```
Test
 public void testWordCountMapper() throws IOException {
mapDriver.withInput(new LongWritable(1), new Text(firstTestKey))
.withInput(new LongWritable(2), new Text(secondTestKey))
.withOutput(new Text(firstTestKey), new IntWritable(1))
.withOutput(new Text("blogspot"), new IntWritable(1))
.runTest();
 }
```

Multiple inputs are supported by MRUnit, as you can see from the code above. You may also initialisefirstTestKey and secondTestKey as String variables in the setUp() function.

The Reducer is tested in the same way.

```
@Test
 public void testWordCountReducer() throws IOException {
    Text firstMapKey = new Text(firstTestKey);
    List<IntWritable>firstMapValues = new ArrayList<IntWritable>();
firstMapValues.add(new IntWritable(1));
firstMapValues.add(new IntWritable(1));

    Text secondMapKey = new Text(secondTestKey);
    List<IntWritable>secondMapValues = new ArrayList<IntWritable>();
secondMapValues.add(new IntWritable(1));
secondMapValues.add(new IntWritable(1));
secondMapValues.add(new IntWritable(1));

reduceDriver.withInput(firstMapKey, firstMapValues)
.withInput(secondMapKey, secondMapValues)
.withOutput(firstMapKey, new IntWritable(2))
.withOutput(secondMapKey, new IntWritable(3))
.runTest();
 }
```

and the overall MapReduce flow

```
@Test
 public void testWordCountMapReducer() throws IOException {
      mapReduceDriver.withInput(new LongWritable(1), new Text(firstTestKey))
.withInput(new LongWritable(2), new Text(firstTestKey))
.withInput(new LongWritable(3), new Text(secondTestKey))
.withOutput(new Text(firstTestKey), new IntWritable(2))
.withOutput(new Text(secondTestKey), new IntWritable(1))
.runTest();
 }
```

The only thing that changes is the precise driver to utilise. Any MRUnit test case may be run in the same manner that JUnit test cases are run. As a result, you may use them all together in your apps. After you've run the command,

mvn test

Maven will execute all available unit tests (both JUnit and MRUnit) for the supplied MapReduce application and create execution results.

10.2 Developing and testing MapReduce jobs with MRUnit

a. **Map Phase:**Conceptually, MapReduce jobs are relatively simple. In the map phase, each input record has a function applied to it, resulting in one or more key-value pairs. The reduce phase receives a group of the key-value pairs and performs some function over that group. Testing mappers and reducers should be as easy as testing any other function. A given input will result in an expected output. The complexities arise due to the distributed nature of Hadoop. Hadoop is a large framework with many moving parts. Prior to the release of MRUnit by Cloudera, even the simplest tests running in local mode would have to read from the disk and take several seconds each to set up and run.

b. **Reduce Phase:**MRUnit removes as much of the Hadoop framework as possible while developing and testing. The focus is narrowed to the map and reduce code, their inputs, and expected outputs. With MRUnit, developing and testing MapReduce code can be done entirely in the IDE, and these tests take fractions of a second to run. This recipe will demonstrate how MRUnit uses the IdentityMapper provided by the MapReduce framework in the lib folder.

The IdentityMapper takes a key-value pair as input and emits the same key-value pair, unchanged.

Step1: Download the latest version of MRUnit from http://mrunit.apache.org/general/downloads.html

Step2: Create a new Java project

Step3: Add the mrunit-X.Y.Z-incubating-hadoop1.jar file and other Hadoop JAR files to the build path of the Java project

Step4: Create a new class named IdentityMapperTest

Step5: For the full source, review the IdentityMapperTest.java file in the source code.

Follow these steps to test a mapper with MRUnit

1. Have the IdentityMapperTest class extends the TestCase class:

   public class IdentityMapperTest extends TestCase

   MRUnit is built on top of the popular JUnit testing framework. It uses the object-mocking library, Mockito, to mock most of the essential Hadoop objects so the user only needs to focus on the map and reduce logic.

2. Create two private members of mapper and driver:

```
private Mapper identityMapper;
private MapDriver mapDriver;
```

The MapDriver class runs the test. It is instantiated with a Mapper class

3. Add a setup() method with a Before annotation:

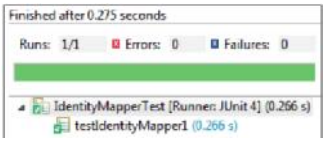
```
@Before
public void setup() { identityMapper = new IdentityMapper(); mapDriver = new MapDriver(identityMapper);
```

4. Add a testIdentityMapper1() method with a Test annotation:

```
@Test
public void testIdentityMapper1() {
mapDriver.withInput(new Text("key"), new Text("value")) mapDriver.withOutput(new Text("key"), new Text("value"))
.runTest();
}
```

The withInput() method is called to provide input to the Mapper class that the MapDriver class was instantiated with. The withOutput() method is called to provide output to validate the results of the call to the Mapper class.

5. Run the application

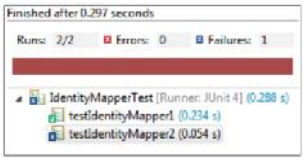


6. Add a testIdentityMapper2() method that would fail:

```
@Test
public void testIdentityMapper2() {
mapDriver.withInput(new Text("key"), new Text("value"))
mapDriver.withOutput(new Text("key2"), new Text("value2"))
mapDriver.runTest();
}
```

The call to the runTest() method actually calls the mapper, passing it the inputs and validating its outputs against the ones provided by the withOutput() method.

7. Run the application again



This example only showed the testing of a mapper. MRUnit also provides a ReduceDriver class that can be used in the same way as MapDriver for testing reducers.

Developing and testing MapReduce jobs running in local mode

Developing in MRUnit and local mode are complementary. MRUnit provides an elegant way to test the map and reduce phases of a MapReduce job. Initial development and testing of jobs should be done using this framework. However, there are several key components of a MapReduce job that are not exercised when running MRUnit tests. Two key class types are InputFormats and OutFormats. Running jobs in local mode will test a larger portion of a job. When testing in local mode, it is also much easier to use a significant amount of real-world data. This recipe will show an example of configuring Hadoop to use local mode and then debugging that job using the Eclipse debugger.

You will need to download the weblog_entries_bad_records.txt dataset from the Packt website, http://www.packtpub.com/support. This example will use the CounterExample.java class provided with the *Using Counters in a MapReduce job to track bad records* recipe.

CounterExample.java class

Using Counters in a MapReduce job to track bad records recipe.

Step1: Open the $HADOOP_HOME/conf/mapred-site.xml file in a text editor.

Step2: The mapred.job.tracker property set to local informs the Hadoop framework that jobs will now run in the local mode. The LocalJobRunner class, which is used when running in local mode, is responsible for implementing the MapReduce framework locally in a single process.

Open the $HADOOP_HOME/conf/mapred-site.xml file in a text editor.

```
<property>
<name>mapred.job.tracker</name>
<value>local</value>
</property>
```

The mapred.job.tracker property set to local informs the Hadoop framework that jobs will now run in the local mode. The LocalJobRunner class, which is used when running in local mode, is responsible for implementing the MapReduce framework locally in a single process. This has the benefit of keeping jobs that run in local mode as close as possible to the jobs that run distributed on a cluster. One downside to using LocalJobRunner is that it carries the baggage of setting up an instance of Hadoop. This means even the smallest jobs will require at least several seconds to run. Setting the fs.default.name property value to file:/// configures the job to look for input and output files on the local filesystem.

Open the $HADOOP_HOME/conf/core-site.xml file in a text editor.

Step3: Set the fs.default.name property value to file:///:

```
<property>
<name>fs.default.name</name>
<value>file:///</value>
</property>
```

Open the $HADOOP_HOME/conf/hadoop-env.sh file and add the following line:

Step4:hadoop-env.sh file and add the following line:

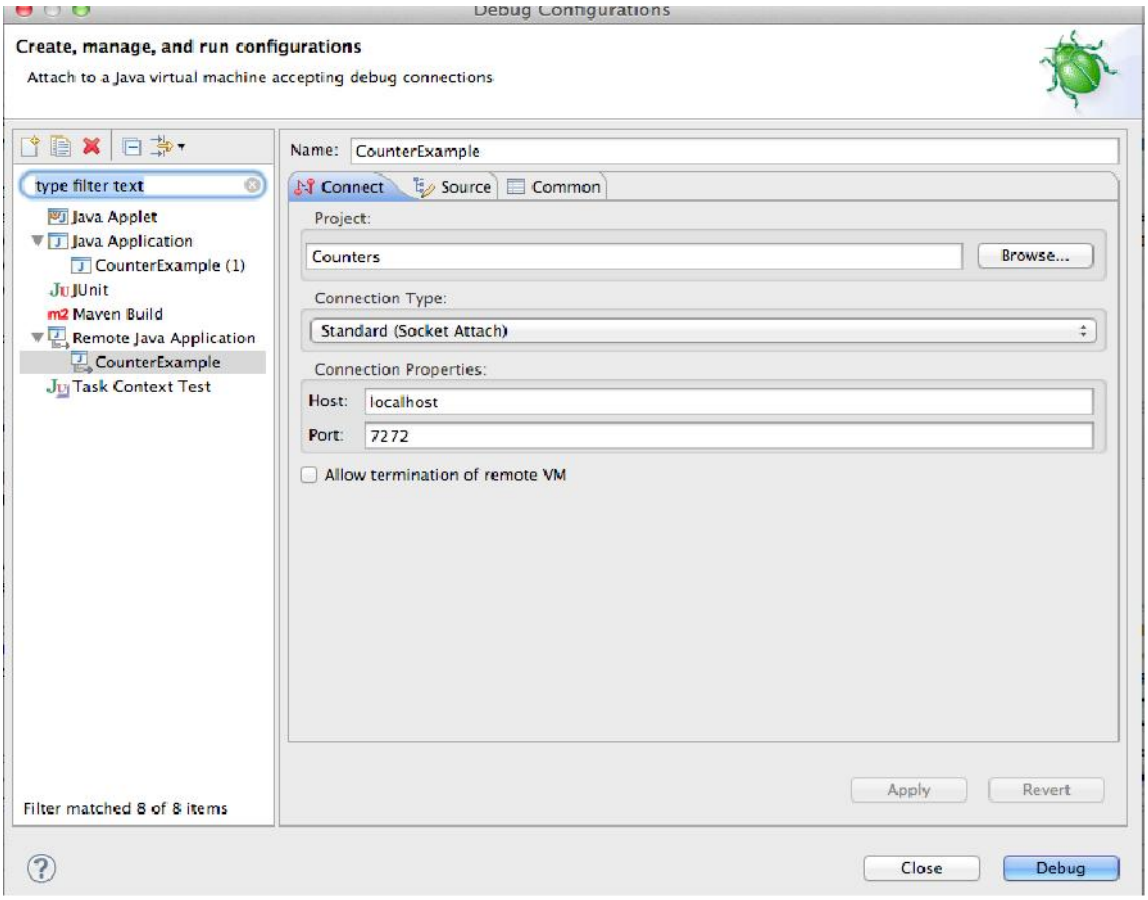
Adding export HADOOP_OPTS="-agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=7272" to the hadoop-env.sh file configures the JVM to suspend processing and listen for a remote debugger on port 7272 on start up.

Run the CountersExample.jar file by passing the local path to the weblog_ entries_bad_records.txt file, and give a local path to an output file:

Step5:$HADOOP_HOME/bin/hadoop jar ./CountersExample.jar com.packt.hadoop.solutions.CounterExample /local/path/to/weblog_entries_bad_ records.txt /local/path/to/weblog_entries_clean.txt

You'll get the following output: Listening for transport dt_socket at address: 7272

Step6:



Step7: Open the Counters project in Eclipse, and set up a new remote debug configuration. Create a new breakpoint and debug.

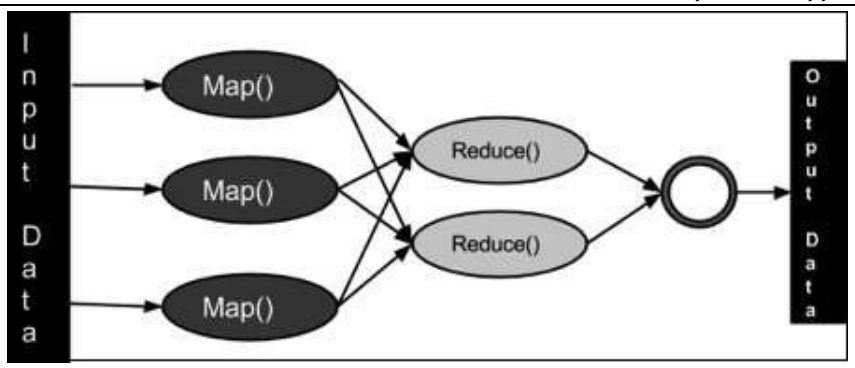
A MapReduce job that is configured to execute in local mode runs entirely in one JVM instance. Unlike the pseudo-distributed mode, this mode makes it possible to hook up a remote debugger to debug a job.

Apache Pig also provides a local mode for development and testing. It uses the same LocalJobRunner class as a local mode MapReduce job. It can be accessed by starting Pig with the following command:

pig –x local

10.3 Learn Anatomy of a MapReduce Job Run

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.



A MapReduce task may be started with only one line of code: JobClient.runJob (conf).

The whole process is illustrated in belowFigure . At the highest level, there are four independent entities:

1. The MapReduce task is submitted by the client.
2. The jobtracker, which keeps track of the job's progress. JobTracker is the primary class of the jobtracker Java programme.
3. Tasktrackers, which carry out the jobs that have been divided up in the work. Tasktrackers are Java apps with TaskTracker as their primary class.
4. The distributed filesystem (DFS), which is utilised to share job files across the various organisations.

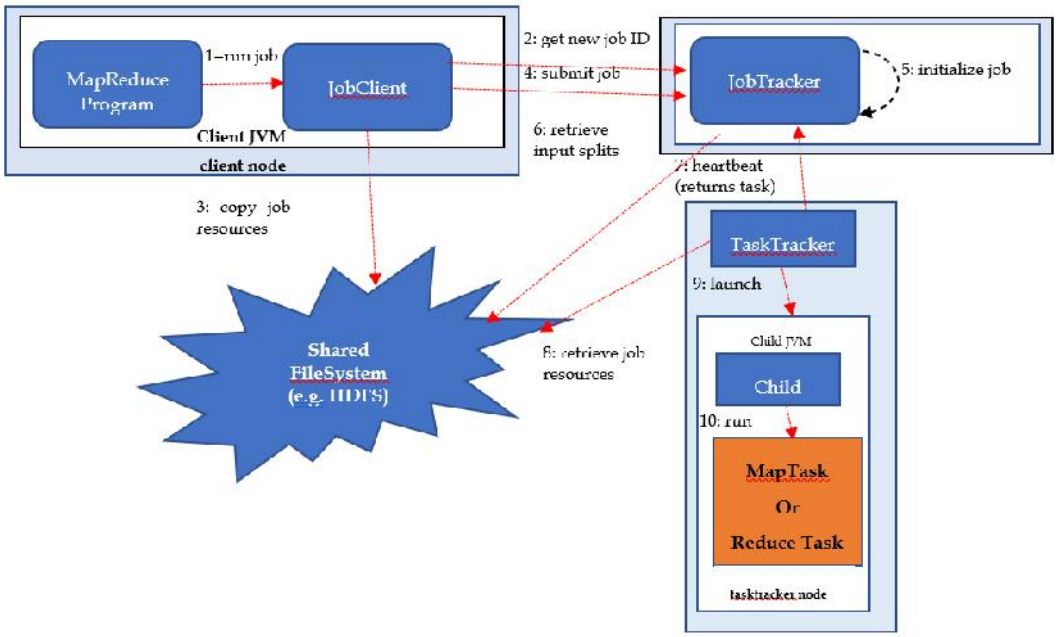


Figure 1 MapReduce Task

The JobClient'srunJob() function generates a new JobClient object and executes submitJob() on it as a convenience method. After submitting the work, runJob() polls the job's progress every second and, if it has changed since the last report, reports it to the console. The job counters are presented when the work is finished and whether it was successful. Otherwise, the console is recorded with the error that caused the job to fail.

The submitJob() function of JobClient implements the following job submission process:

- Requests a new job ID from the jobtracker (through getNewJobId() on JobTracker) (step 2).

- Checks the job's output specification. The task is not submitted if the output directory is not given or if it already exists, for example, and an error is issued to the MapReduce application.
- Calculates the job's input splits. The task is not submitted and an error is issued to the MapReduce software if the splits cannot be computed, for example because the input pathways do not exist.
- Copies the resources needed to perform the task to the jobtracker's filesystem in a directory named after the job ID, including the job JAR file, the configuration file, and the computed input splits. The job JAR is duplicated with a high replication factor (set by the mapred.submit.replication parameter, which defaults to 10) so that tasktrackers may access many copies when running tasks for the job.
- By invoking submitJob() on JobTracker, tells the jobtracker that the job is ready to be executed.

Job Initialization

When the JobTracker gets a call to its submitJob() function, it places it in an internal queue, where it will be picked up and initialised by the job scheduler. Initialization entails constructing an object to represent the job being executed, which encapsulates its activities, as well as accounting information to track the state and progress of the tasks (step 5).

The job scheduler gets the input splits computed by the JobClient from the shared filesystem before creating the list of jobs to perform (step 6). For each split, it then produces a separate map task. The scheduler simply produces this amount of reduction tasks to perform, which is defined by the mapred.reduce.tasks property in the JobConf, which is set by the setNumReduce Tasks() function. At this moment, task IDs are assigned.

Task Assignment

Tasktrackers use a simple loop to deliver heartbeat method calls to the jobtracker on a regular basis. Heartbeats not only inform the jobtracker that a tasktracker is alive, but they also serve as a messaging channel. A tasktracker will signal if it is ready to run a new task as part of the heartbeat, and if it is, the jobtracker will assign it a task, which it will send to the tasktracker using the heartbeat return value (step 7).

The jobtracker must choose a job from which to select a work for the tasktracker before it may choose a task for the tasktracker. There are other scheduling methods, as detailed later in this chapter (see "Work Scheduling"), but the default one merely keeps track of job priorities. The jobtracker now selects a task for the job after deciding on a job.

For map tasks and reduce tasks, tasktrackers have a set number of slots: for example, a tasktracker may be able to execute two map tasks and two reduce tasks at the same time. (The exact number is determined on the number of cores and memory available on the tasktracker; see "Memory") Because the default scheduler fills empty map task slots before reduce task slots, the jobtracker will choose a map task if the tasktracker has at least one empty map task slot; otherwise, it will select a reduce task.

Because there are no data locality constraints, the jobtracker simply chooses the next reduce task from its list of yet-to-be-run reduce jobs. For a map job, however, it considers the tasktracker's network position and selects a task with an input split that is as close to the tasktracker as feasible. In the best case scenario, the job is data-local, meaning it runs on the same node as the split. Alternatively, the job might be rack-local, meaning it's on the same rack as the split but not on the same node. Some jobs are neither data-local nor rack-local, and their data is retrieved from a rack other than the one on which they are operating. Looking at a job's counters will reveal the proportion of each sort of work.

Task Execution

After the tasktracker has been given a task to complete, the next step is for it to complete the task. First, it copies the job JAR from the common filesystem to the tasktracker's filesystem to localise it. It also moves any files required by the programme from the distributed cache to the local disc (see "Distributed Cache") (step 8). Second, it creates a task-specific local working directory and un-jars the JAR's contents into it. Third, it creates a TaskRunner object to carry out the task.

TaskRunner creates a new Java Virtual Machine (step 9) to perform each task in (step 10), ensuring that any faults in the user-defined map and reduce routines have no impact on the tasktracker (by causing it to crash or hang, for example). However, the JVM can be reused across jobs (see "Task JVM Reuse").

The umbilical interface is used by the child process to interact with its parent. This manner, it keeps the parent updated on the task's progress every few seconds until it's finished.

Streaming and Pipes

Both Streaming and Pipes perform unique map and reduce actions in order to launch and communicate with the user-supplied executable (Figure).

In the instance of Streaming, the Streaming task uses conventional input and output streams to interact with the process (which can be written in any language). The Pipes job, on the other hand, listens on a socket and sends a port number to the C++ process in its environment, allowing the C++ process to establish a permanent socket connection back to the parent Java Pipes task on startup.

The Java process delivers input key-value pairs to the external process during task execution in both circumstances, which runs them through the user-defined map or reduce function and returns the output key-value pairs to the Java process. From the tasktracker's perspective, it's as though the map or reduce code was executed by the tasktracker child process.

Progress and Status Updates

MapReduce jobs are long-running batch operations that might take anything from a few minutes to many hours to complete. Because this will take a long time, it's critical that the user get feedback on how the work is proceeding. A job's status, as well as the status of each of its tasks, contains information such as the job's or task's current state (e.g., running, successfully finished, failed), the progress of maps and reduces, the values of the job's counters, and a statusmessage or description (which may be set by user code). How do these statuses get conveyed back to the customer as they change throughout the job? When a job is running, it tracks its progress, or the percentage of the work that has been accomplished. This is the percentage of the input that has been processed for map tasks.

It's a little more complicated for decrease jobs, but the system can still estimate the proportion of the reduction input handled. It accomplishes this by separating the entire progress into three pieces, each of which corresponds to one of the three shuffle stages (see "Shuffle and Sort"). For example, if the task has finished the copy and sort phases (1/3) and is halfway through the reduce phase (1/6) and has run the reducer on half of its input, the work's progress is (5/6).

Job Completion

The status of a job is changed to "successful" when the jobtracker receives news that the last task for the job has been completed. The JobClient then learns that the work has completed successfully when it polls for status, so it produces a message to inform the user and then exits the runJob() function.

If the jobtracker is enabled to do so, it will additionally send an HTTP job notice. Clients that want to get callbacks can set this up using the job.end.notifica tion.url parameter. Finally, the jobtracker cleans up its working environment for the job, and tasktrackers are instructed to do the same (so intermediate output is deleted, for example).

10.4 Hadoop Scheduler

Prior to Hadoop 2, Hadoop MapReduce was a software framework for developing applications that process enormous volumes of data in parallel on a large Hadoop cluster (terabytes to petabytes). This framework is in charge of task scheduling, monitoring, and re-execution when a task fails.

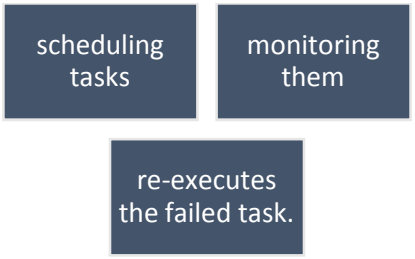


Figure 2 Hadoop MapReduce is in charge of

In Hadoop 2, a **YARN** called Yet Another Resource Negotiator was introduced. The basic idea behind the YARN introduction is to split the functionalities of resource management and job scheduling or monitoring into separate daemons that are ResorceManager, ApplicationMaster, and NodeManager. ResourceManager is the master daemon that arbitrates resources among all the applications in the system. NodeManager is the slave daemon responsible for containers, monitoring their resource usage, and reporting the same to ResourceManager or Schedulers. ApplicationMaster negotiates resources from the ResourceManager and works with NodeManager in order to execute and monitor the task.The ResourceManager has two main components that are Schedulers and ApplicationsManager as shown in **Error! Reference source not found.**.

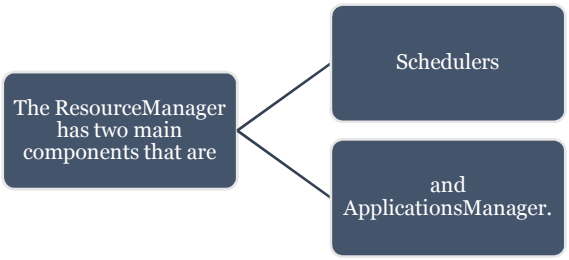


Figure 3 Components of resource manager

Schedulers in YARN

ResourceManager is a pureschedulerwhich is responsible for allocating resources to the various running applications.It is not responsible for monitoring or tracking the status of an application. Also, the scheduler does not guarantee about restarting the tasks that are failed either due to hardware failure or application failure.

The scheduler performs scheduling based on the resource requirements of the applications. It has some pluggable policies that are responsible for partitioning the cluster resources among the various queues, applications, etc. The FIFO Scheduler, CapacityScheduler, and FairScheduler are such pluggable policies that are responsible for allocating resources to the applications.

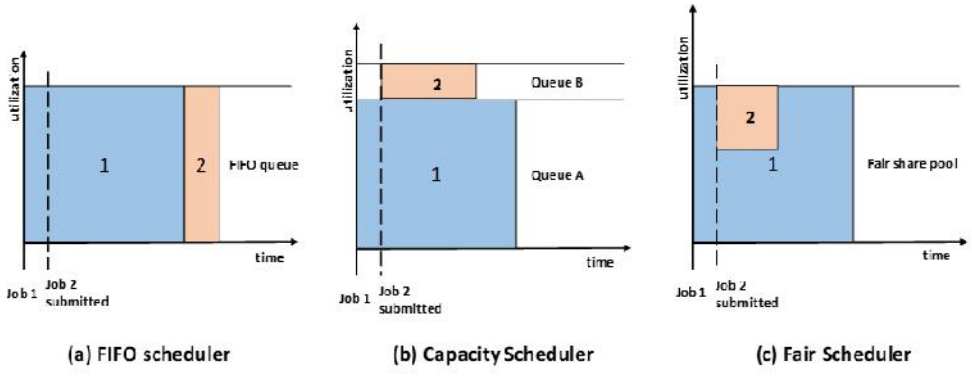


Figure 4 Types of Hadoop Scheduler

FIFO Scheduler

First In First Out is the default scheduling policy used in Hadoop. FIFO Scheduler gives more preferences to the application coming first than those coming later. It places the applications in a queue and executes them in the order of their submission (first in, first out).

Here, irrespective of the size and priority, the request for the first application in the queue are allocated first. Once the first application request is satisfied, then only the next application in the queue is served.

Advantages of FIFO Scheduling

- It is simple to understand and doesn't need any configuration.
- Jobs are executed in the order of their submission.

Disadvantages of FIFO Scheduling

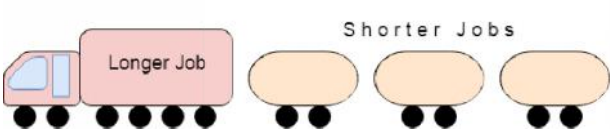


Figure 5 FIFO Scheduling

It is not suitable for shared clusters. If the large application comes before the shorter one, then the large application will use all the resources in the cluster, and the shorter application has to wait for its turn. This leads to starvation.It does not take into account the balance of resource allocation between the long applications and short applications.

Capacity Scheduler

- The CapacityScheduler allows multiple-tenants to securely share a large Hadoop cluster. It is designed to run Hadoop applications in a shared, multi-tenant cluster while maximizing the throughput and the utilization of the cluster. It supports hierarchical queues to reflect the structure of organizations or groups that utilizes the cluster resources. A queue hierarchy contains three types of queues that are root, parent, and leaf. The root queue represents the cluster itself, parent queue represents organization/group or sub-organization/sub-group, and the leaf accepts application submission. Also, when there is a demand for the free resources that are available on the queue who has completed its task, by the queues running below capacity, then these resources will be assigned to the applications on queues running below capacity. This provides elasticity for the organization in a cost-effective manner. Apart from it, the CapacityScheduler provides a comprehensive set of limits to ensure that a single application/user/queue cannot use a disproportionate amount of resources in the cluster.

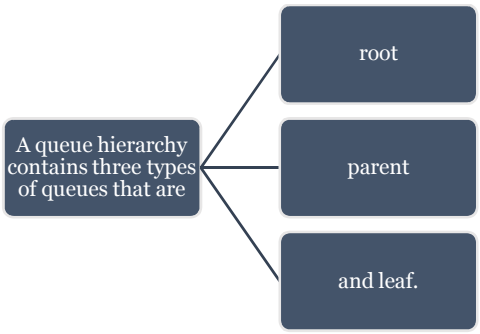


Figure 6 Types of Queues

To ensure fairness and stability, it also provides limits on initialized and pending apps from a single user and queue.It maximizes the utilization of resources and throughput in the Hadoop cluster. Provides elasticity for groups or organizations in a cost-effective manner.It also gives capacity guarantees and safeguards to the organization utilizing cluster.It is complex amongst the other scheduler.

Fair Scheduler

FairScheduler allows YARN applications to fairly share resources in large Hadoop clusters. With FairScheduler, there is no need for reserving a set amount of capacity because it will dynamically balance resources between all running applications. It assigns resources to applications in such a way that all applications get, on average, an equal amount of resources over time. The FairScheduler, by default, takes scheduling fairness decisions only on the basis of memory. We can configure it to schedule with both memory and CPU.When the single application is running, then that app uses the entire cluster resources. When other applications are submitted, the free up resources are assigned to the new apps so that every app eventually gets roughly the same amount of resources. FairScheduler enables short apps to finish in a reasonable time without starving the long-lived apps.Similar to CapacityScheduler, the FairScheduler supports hierarchical queue to reflect the structure of the long shared cluster.Apart from fair scheduling, the FairScheduler allows for assigning minimum shares to queues for ensuring that certain users, production, or group applications always get sufficient resources. When an app is present in the queue, then the app gets its minimum share, but when the queue doesn't need its full guaranteed share, then the excess share is split between other running applications.

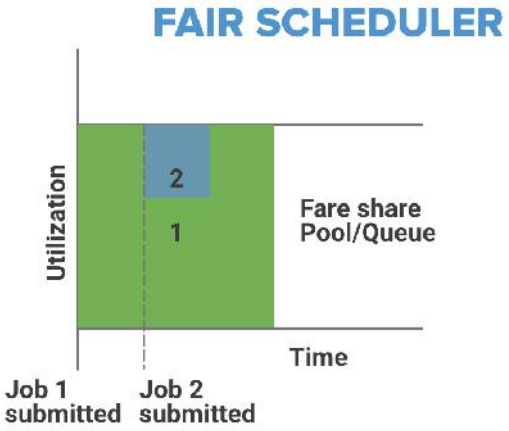


Figure 7 Fair Scheduler

Advantages of fair scheduler

It provides a reasonable way to share the Hadoop Cluster between the number of users. Also, the FairScheduler can work with app priorities where the priorities are used as weights in determining the fraction of the total resources that each application should get.

Disadvantages of Fair scheduler

- It requires configuration.
- Different options of pluggable scheduling policies like FIFO, FairScheduler, and CapacityScheduler provided by Hadoop YARN ResourceManager for scheduling resources amongst the multiple applications running in the Hadoop cluster.

10.5 Failures in MapReduce

In classic MapReduce, there can be three kinds of failures

a. Failure of map reduce task

b. Failure of task tacker

c. Failure of Job Tracker

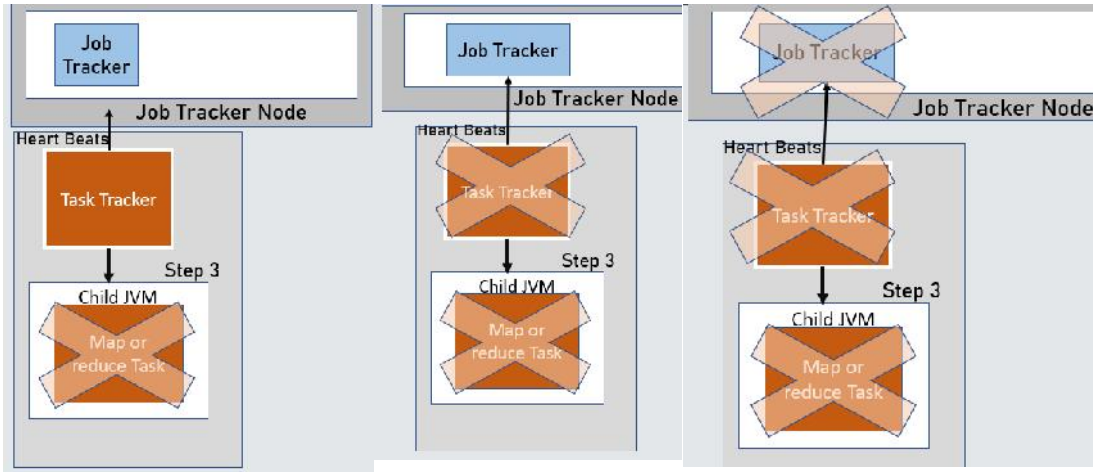


Figure 8 Three Kinds of Failure

Failures of MapReduce task

We are going to discuss all the cases one by one. Task are the user codes. There can be scenario where user code may be running to infinite loop. In these cases task tracker will observe there has not been any progress on the task for period of time. Then, it would mark a job failed. The observation time set by property "**mapred.task.timeout**"

It can be set to 0 as well. In that case, task tracker would never fail a long running job. This is not suggested, As this slots would not free up quickly if the task is stuck. This would bring down the overall performance of the clusters.Another reason of failure of user task can be run time errors. In that case, error is reported back to task tracker. And task tracker would put it in the user log in that scenario.

1. There can be rear case where JVM may have been exposed to a bug. In that case, a task JVM can crash along with task tracker. In those cases job tracker notices, child jvm has exited and marks the task has failed. All the failed task attempts are notified to jobtracker.
2. And job tracker reschedules the execution of failed task onto a different task tracker. This is done so as to ensure the reason of failure is underline hardware
3. The number of reattempts that could be made on map task is governed by the properties mapred.map.max.attempts=4(default)
4. Similarly for reduce task is governed by mapred.reduce.max.attempts=4(default)

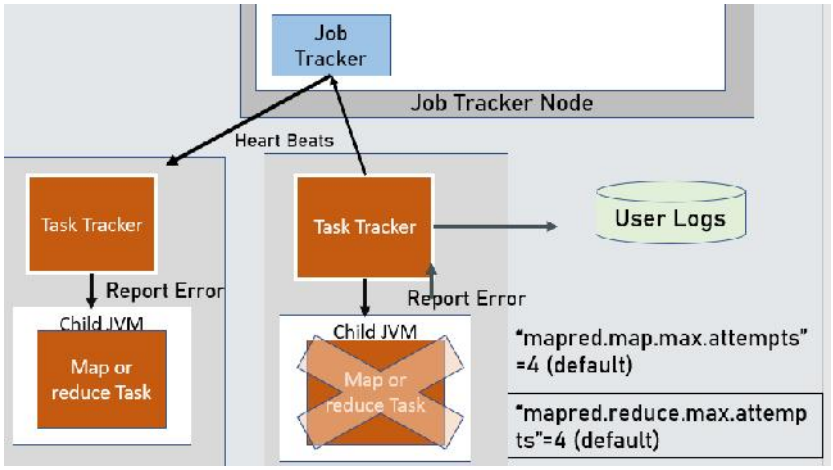


Figure 9 Failures in Classic MapReduce

Failure of Task Tracker

- In that case the job tracker stops receiving the heartbeats from the task tracker. Thus, the job tracker concludes that the task tracker is dead. In that case it reschedules the task on another task tracker. Job Tracker reschedules the task which didn't complete and the task which got completed but its job didn't complete. Even the completed task is rerun as the results would be return to local disk and they would have been lost due to crash of the task tracker.As soon as the Job tracker realizes that the heart beat from the task tracker is stopped. The job tracker removes the task tracker from its available pool of task tracker. That is not the only task criteria on which the task tracker can be removed from the available pool.
- If the number of task failed on the task tracker crosses threshold . It gets blocklisted and removed from the available pool of task tracker. The threshold is set by the property mapred.max.tracker.failures.
- In case if the task tracker is blacklisted, it joins back on the restart or after the certain period of time.

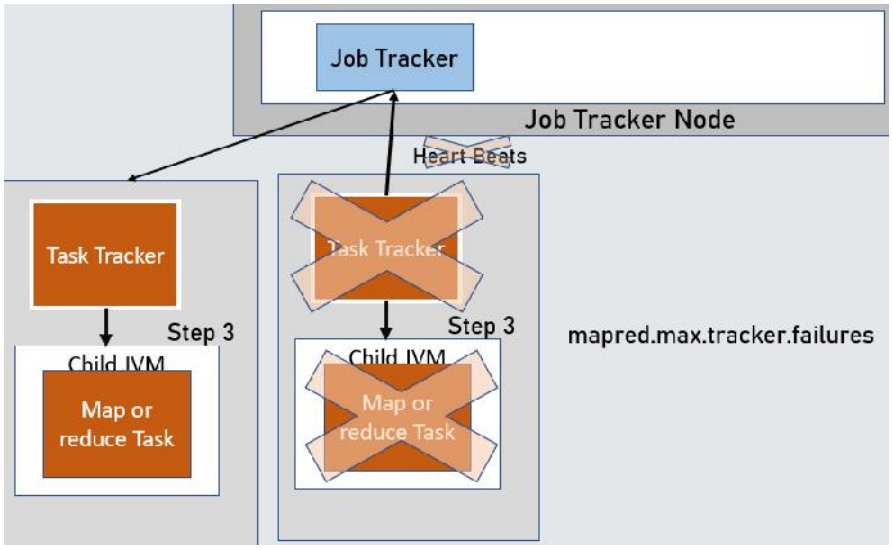


Figure 10 Failures of Task Tracker

Failure of JobTracker

- The final case can be job trackerfailure. It is the most serious failure in classic mapreduce. Nothing much can be dine in this case. Job tracker is single point failure in MapReduce.
- So it is recommended to be run on the better hardware so as to avoid the scenario as much as possible. We need to resubmit all the jobs in progress once the jobtracker is brought up again. In YARN,this situation is little improved.

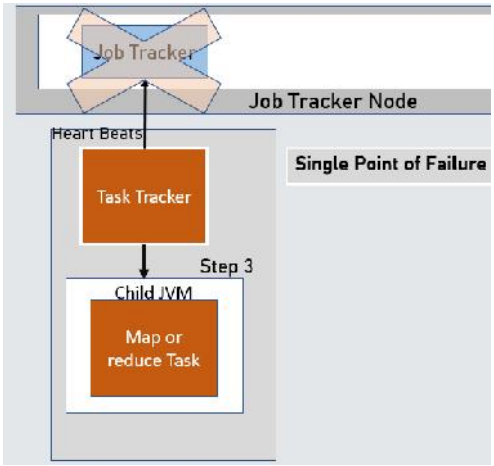


Figure 11Failures of JobTracker

Failure Cases-Yarn

- In the real world, user code is *buggy*, processes *crash*, and machines *fail*. One of the major benefits of using Hadoop is its ability to handle such *failures* and allow your job to complete.
- **Failures in YARN:** For MapReduce programs running on YARN, we need to consider the failure of any of the following entities: the *task*, the *application master*, the *node manager*, and the *resource manager as shown in* Figure 12.

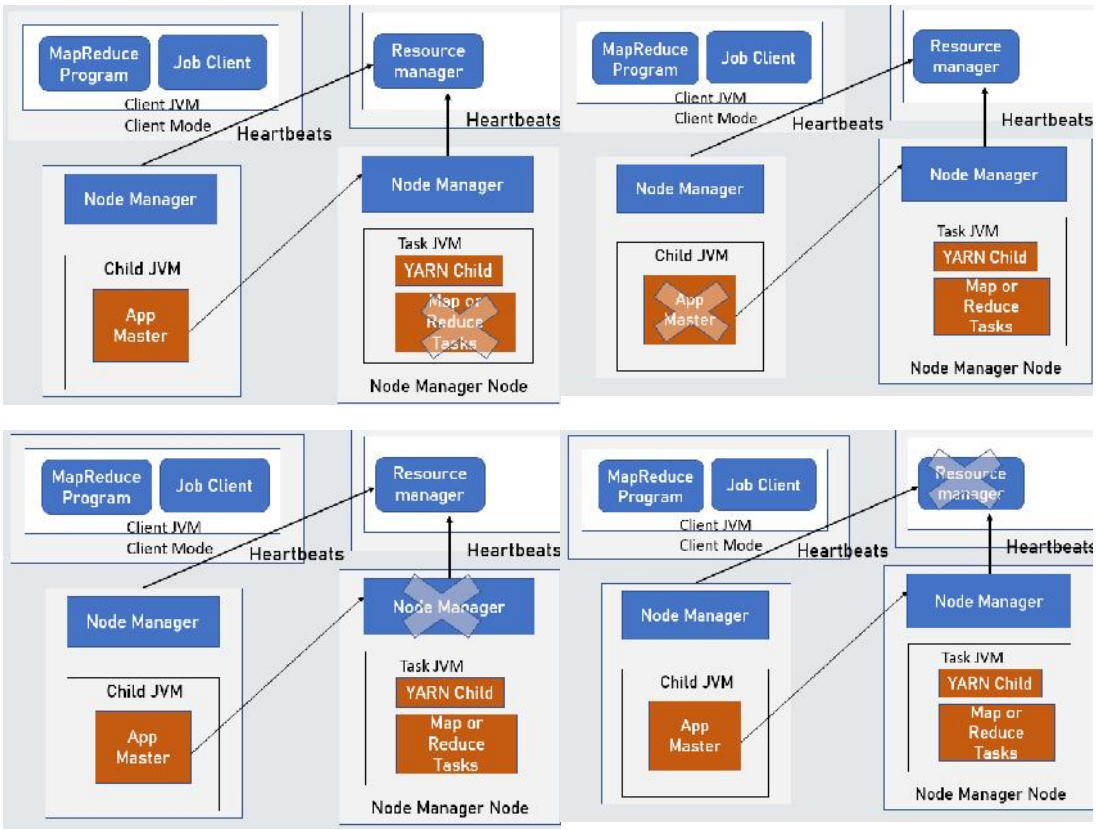


Figure 12 Cases of failures in Yarn

- **Task Failure**

  Failure of the running task is similar to the classic case. Runtime exceptions and sudden exits of the JVM are propagated back to the application master and the task attempt is marked as failed. Likewise, hanging tasks are noticed by the application master by the absence of a ping over the umbilical channel (the timeout is set by mapreduce.task.time out), and again the task attempt is marked as failed.The configuration properties for determining when a task is considered to be failed are the same as the classic case: a task is marked as failed after four attempts (set by mapreduce.map.maxattempts for map tasks and mapreduce.reduce.maxattempts for reducer tasks).

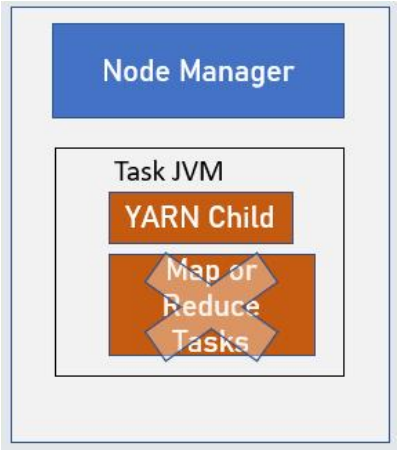


Figure 13 Task failure

A job will be failed if more than mapreduce.map.failures.maxpercent percent of the map tasks in the job fail, or more than mapreduce.reduce.failures.maxpercent percent of the reduce tasks fail.

Application Master Failure

Just like MapReduce tasks are given several attempts to succeed (in the face of hardware or network failures) applications in YARN are tried multiple times in the event of failure. By default, applications are marked as failed if they fail once, but this can be increased by setting the property yarn.resourcemanager.am.max-retries.

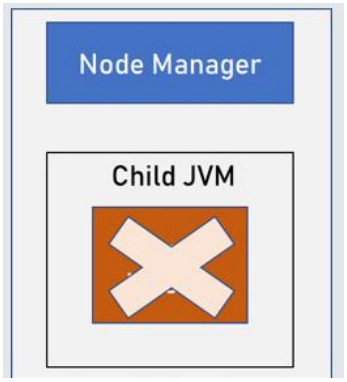


Figure 14 Application master failure

An application master sends periodic heartbeats to the resource manager, and in the event of application master failure, the resource manager will detect the failure and start a new instance of the master running in a new container (managed by a node manager)

In the case of the MapReduce application master, it can recover the state of the tasks that had already been run by the (failed) application so they don't have to be rerun. By default, recovery is not enabled, so failed application masters will not rerun all their tasks, but you can turn it on by setting yarn.app.mapreduce.am.job.recovery.enable to true.

The client polls the application master for progress reports, so if its application master fails the client needs to locate the new instance.

During job initialization the client asks the resource manager for the application master's address, and then caches it, so it doesn't overload the the resource manager with a request every time it needs to poll the application master.

If the application master fails, however, the client will experience a timeout when it issues a status update, at which point the client will go back to the resource manager to ask for the new application master's address.

Node Manager Failure

- If a node manager fails, then it will stop sending heartbeats to the resource manager, and the node manager will be removed from the resource manager's pool of available nodes.

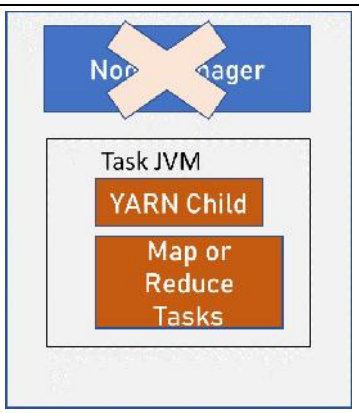


Figure 15 Node Manager Failure

- The property yarn.resourcemanager.nm.liveness-monitor.expiry-intervalms, which defaults to 600000 (10 minutes), determines the minimum time the resource manager waits before considering a node manager that has sent no heartbeat in that time as failed.
- Any task or application master running on the failed node manager will be recovered using the mechanisms described in the previous two sections.
- Node managers may be blacklisted if the number of failures for the application is high. Blacklisting is done by the application master, and for MapReduce the application master will try to reschedule tasks on different nodes if more than three tasks fail on a node manager.
- The threshold may be set with
- mapreduce.job.maxtaskfailures.per.tracker.

Resource Manager Failure

Failure of the resource manager is serious, since without it neither jobs nor task containers can be launched. The resource manager was designed from the outset to be able to recover from crashes, by using a checkpointing mechanism to save its state to persistent storage, although at the time of writing the latest release did not have a complete implementation.

After a crash, a new resource manager instance is brought up (by an adminstrator) and it recovers from the saved state. The state consists of the node managers in the system as well as the running applications.

(Note that tasks are not part of the resource manager's state, since they are managed by the application. Thus the amount of state to be stored is much more managable than that of the jobtracker.)

The storage used by the resource manager is configurable via the yarn.resourcemanager.store.class property. The default is org.apache.hadoop.yarn.server.resourcemanager.recovery.MemStore, which keeps the store in memory, and is therefore not highly-available.

However, there is a ZooKeeper-based store in the works that will support reliable recovery from resource manager failures in the future.

10.6 Shuffling

In Hadoop, the process by which the intermediate output from mappers is transferred to the reducer is called Shuffling. Reducer gets 1 or more keys and associated values on the basis of reducers. Intermediated key-value generated by mapper is sorted automatically by key.

What is Shuffling and Sorting in Hadoop MapReduce?

Before we start with Shuffle and Sort in MapReduce, let us revise the other phases of MapReduce like Mapper, reducer in MapReduce, Combiner, partitioner in MapReduce and inputFormat in MapReduce.

Figure 16 Phases of MapReduce

Mapper task is the first phase of processing that processes each input record (from RecordReader) and generates an intermediate key-value pair. **Hadoop** Mapper store intermediate-output on the local disk. In this Hadoop mapper tutorial, we will try to answer what is a **MapReduce** Mapper how to generate **key-value pair** in Hadoop, what is InputSplit and RecordReader in Hadoop, how mapper works in Hadoop.

Reducer in MapReduce

Reducer in Hadoop MapReduce **reduces a set of intermediate values which share a key to a smaller set of values**. In MapReduce job execution flow, Reducer takes a set of an intermediate key-value pair produced by the mapper as the input.In Hadoop, **Reducer** takes the output of the Mapper (intermediate key-value pair) process each of them to generate the output. The output of the reducer is the final output, which is stored in HDFS. Usually, in the Hadoop Reducer, we do aggregation or summation sort of computation.

Combiner

MapReduce - Combiners. A Combiner, also known as a semi-reducer, is **an optional class that operates by accepting the inputs from the Map class** and thereafter passing the output key-value pairs to the Reducer class. The main function of a Combiner is to summarize the map output records with the same key.

Partitioner in MapReduce

The Partitioner in MapReduce **controls the partitioning of the key of the intermediate mapper output**. By hash function, key (or a subset of the key) is used to derive the partition. A total number of partitions depends on the number of reduce task.

InputFormat in MapReduce

Hadoop can process many different types of data formats, from flat text files to databases. In this section, we will explore the different formats available in next chapter

Working of Shuffle and Sort.

Shuffle and sort steps which are core and hard to every MapReduce job. Every mapreduce goes through a shuffle and sort phase.

Map:- process input key and value then map output id sorted and is transferred to reducer and that is known as shuffle.

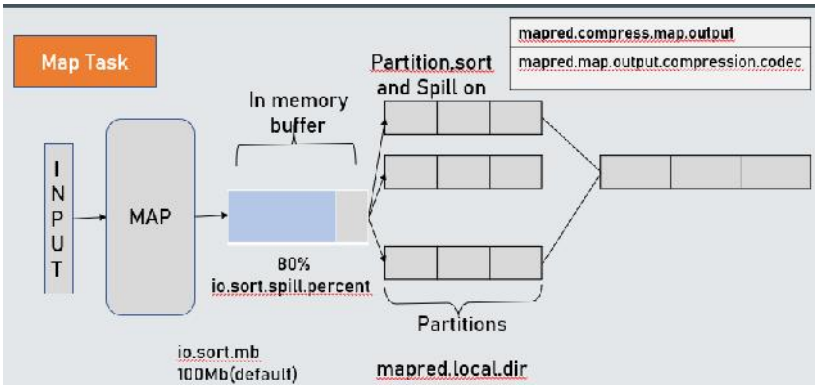


Figure 17 Working of shuffle and sort

1)The input

2) Output is not directly written to disk but is written to memory buffer.

3) Size of the buffer is decided by the property io.sort.mb.

4)Default size is 100MB

5) If map has more of output, it may fill up the buffer and in that case map would be paused for a while till the spill empties the buffer.

6) After spill completed of map may again reach to the threshold.

7)In that case,another spill would be writte in round robin fashion.

8) These are written to the directly specified in the property map.local.dir

9) So there can be many spills before the last key value pair has been written by the map tasks.

10) Each spill is partitioned and sorted by the key and this run through a combiner, if the combiner function is designed for the job.

11) Once map has finished to process all the records. All the splits are merged to an output file which is partitioned and sorted.

12) If more than 3 splits are merged together, combiner function is again run through the final output. 13) Remember that the combiner functions can run many times without changing the final results.

14) Combiner function reduces the size of output which is advantages as there will be less amount of data that would required to be transferred to the reducer machine.

15) If the maps output is going to be very large it is recommended to compress the maps output to reduce the amount of data.

16) This can be done by setting up the property method mapred.compress.map.output to true.

And compression scheme can be specified by the property. Mapred.map.output.compression.codec

18) After this comes the copy phase there would be many map tasks running and they may finish at different times.

19) As soon as they finish, they notified the job tracker or the app master which asks the reducer to copy the result to the local disk and so the partitions are copied by the reducer from the network.

- After this comes the sort phase, reducer merges the map output which are then filled into reducer to create the final result. The mechanism in sort phase is little more involved.Let's look at the sort phase the property which plays an important role is merge factor. io.sort.factor. Its default value is 10. It signifies how many files can be merge at one go.Suppose if Reducer receives 30 files from different maps. then these can be merged into batches of 10 and in three rounds it will create the intermediate merge file and in the final round it would be fed directly into the reducer. Just note that the most files need to be sorted by the keys as well. To increase disk i/o efficiency, the actual algorithms behave differently. It picks first three files and merge into one and then picks up the next patches of ten. In the final round it would take the remaining six files. Doing like this, increases the I/O efficiency.

Objective

In **Hadoop**, the process by which the intermediate output from **mappers** is transferred to the **reducer** is called Shuffling. Reducer gets 1 or more keys and associated values on the basis of reducers. Intermediated **key-value** generated by mapper is sorted automatically by key. In this blog, we will discuss in detail about shuffling and Sorting in **Hadoop MapReduce**.Shuffle phase in Hadoop transfers the map output from Mapper to a Reducer in MapReduce. Sort phase in MapReduce covers the merging and sorting of map outputs. Data from the mapper are grouped by the key, split among reducers, and sorted by the key. Every reducer obtains all values associated with the same key. Shuffle and sort phase in Hadoop occur simultaneously and are done by the MapReduce framework. **Shuffle phase** in Hadoop transfers the map output from Mapper to a Reducer in MapReduce. **Sort phase** in MapReduce covers the merging and sorting of map outputs. Data from the mapper are grouped by the key, split among reducers and sorted by the key. Every reducer obtains all values associated with the same key. Shuffle and sort phase in Hadoop occur simultaneously and are done by the MapReduce framework.

Shuffling in MapReduce

The process of transferring data from the mappers to reducers is known as shuffling i.e. the process by which the system performs the sort and transfers the map output to the reducer as input. So, MapReduce shuffle phase is necessary for the reducers, otherwise, they would not have any input (or input from every mapper). As shuffling can start even before the map phase has finished so this saves some time and completes the tasks in lesser time.

Sorting in MapReduce

The keys generated by the mapper are automatically sorted by MapReduce Framework, i.e. Before starting of reducer, all intermediate **key-value pairs** in MapReduce that are generated by mapper get sorted by key and not by value. Values passed to each reducer are not sorted; they can be in any order.Sorting in Hadoop helps reducer to easily distinguish when a new reduce task should start. This saves time for the reducer. Reducer starts a new reduce task when the next key in the sorted input data is different than the previous. Each reduce task takes key-value pairs as input and generates key-value pair as output.Note that shuffling and sorting in Hadoop MapReduce is not performed at all if you specify zero reducers (setNumReduceTasks(0)). Then, the MapReduce job stops at the map phase, and the map phase does not include any kind of sorting (so even the map phase is faster).

Secondary Sorting in MapReduce

If we want to sort reducer's values, then the secondary sorting technique is used as it enables us to sort the values (in ascending or descending order) passed to each reducer.In conclusion, Shuffling-Sorting occurs simultaneously to summarize the Mapper intermediate output. Shuffling and sorting in Hadoop MapReduce are not performed at all if you specify zero reducers (setNumReduceTasks(0)).The framework sorts all intermediate key-value pairs by key, not by value. It uses secondary sorting for sorting by value.

Summary

- Hadoop The unit tests are all built to run on a single computer rather than a Hadoop cluster. Apache Bigtop has a running search for that. The unit tests function by generating miniDFS, MiniYARN, and MiniMR clusters, as needed. All of them execute the code for the respective services.
- Hadoop has been utilising JUnit4 for some time, yet it appears that many new tests for JUnit v3 are still being produced.
- Apache MRUnit TM is a Java package for unit testing Apache Hadoop map reduce tasks. The post's example uses the Weather dataset, and it works with the year and temperature retrieved from it. Obviously, you can simply adapt the example to your own data.
- Hadoop includes a RecordReader that transforms input splits into key-value pairs using TextInputFormat. In the mapping process, the key-value pairs are utilised as inputs. The only data format that a mapper can read and understand is this one.

- Hadoop applications use the Hadoop Distributed File Solution (HDFS) as their primary data storage system. HDFS is a distributed file system that uses a NameNode and DataNode architecture to allow high-performance data access across highly scalable Hadoop clusters.
- Hadoop is an Apache Foundation open source framework for processing huge volumes of heterogeneous data sets in a distributed way across clusters of commodity computers and hardware using a simplified programming style. Hadoop is a dependable distributed storage and analysis solution.
- If a job fails, Hadoop will identify the failure and reschedule replacements on healthy computers. It will only end the task if it fails four times, which is the default setting that may be changed, and it will kill terminate the job. to be finished
- Namenode stores information about all other nodes in the Hadoop Cluster, files in the cluster, file component blocks and their positions in the cluster, and other information that is necessary for the Hadoop Cluster's functioning.
- Job Tracker manages the sharing of information and outcomes by keeping track of the specific tasks/jobs allocated to each of the nodes.
- The CapacityScheduler was created to allow huge clusters to be shared while ensuring that each organisation has a minimum capacity guarantee. The key principle is that the Hadoop Map-Reduce cluster's available resources are partitioned among various companies that finance the cluster collectively based on computation demands.

Keywords

Apache MRUnit: Apache MRUnit TM is a Java package for unit testing Apache Hadoop map reduce tasks. The post's example uses the Weather dataset, and it works with the year and temperature retrieved from it. Obviously, you can simply adapt the example to your own data.

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

Job Tracker: The job tracker is a master daemon that runs on the same node as the data nodes and manages all of the jobs. This data will be stored on multiple data nodes, but it is the task tracker's responsibility to keep track of it.

FIFO: As the name implies, FIFO stands for First In First Out, which means that the tasks or applications that arrive first are served first. In Hadoop, this is the default Schedular. The jobs are placed in a queue and completed in the order in which they were submitted.

Capacity Scheduler: The CapacityScheduler was created to allow huge clusters to be shared while ensuring that each organisation has a minimum capacity guarantee. The key principle is that the Hadoop Map-Reduce cluster's available resources are partitioned among various companies that finance the cluster collectively based on computation demands.

Fair scheduling is a method of allocating resources to apps in such a way that each app receives an equal proportion of resources over time. Hadoop NextGen can schedule a variety of resource kinds. The Fair Scheduler's scheduling fairness judgments are based only on memory by default.

Task Failure: If a job fails, Hadoop will identify the failure and reschedule replacements on healthy computers. It will only end the task if it fails four times, which is the default setting that may be changed, and it will kill terminate the job. to finalise

Child JVM: The parent MRAppMaster's environment is passed down to the child job. Themapreduce.map.java.optsandmapred.reduce.java.opts configuration arguments in theJob can be used to provide the child JVM extra options. -Djava.library.path=>, for example, can be used to specify non-standard paths for the runtime linker to look for shared libraries. If the symbol(taskid) property is present in the mapreduce.map.java.opts or mapred.reduce.java.opts properties, it is interpolated with the taskid value of the MapReduce task.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Name node: The name node is a piece of commodity hardware that houses the GNU/Linux operating system as well as name node software. It's a piece of software that can run on standard hardware.

Shuffling and sorting: The process of transferring intermediate output from the mapper to the reducer is known as shuffle. On the basis of reducers, a reducer receives one or more keys and their associated values. The mapper's intermediated key – value is automatically ordered by key.

Self Assessment

1. Testing the entire system's end-to-end functioning is characterized as

A. Functional testing

B. Unit Testing

C. Stress Testing

D. Load Testing

2. What is testing?

A. Finding broken code.

B. Evaluating deliverable to find errors.

C. A stage of all projects.

D. All of the above.

3. Which of the following are unit testing techniques.

A. Black box testing

B. White box testing

C. Gray box testing

D. All of the above

4. _____ is a unit testing framework for the Java programming language.

A. JUnit

B. MRUnit

C. Javaunit

D. TestUnit

5. Which of the following are MRUnit core classes.

A. MapDriver

B. ReduceDriver

C. MapReduceDriver

D. All of the above

6. _____ is a processing technique and a program model for distributed computing based on java.

A. Composing

B. Decomposing

C. MapReduce

D. None of above

7. _____ a data processing application into *mappers* and *reducers* is sometimes nontrivial.

A. Composing
B. Decomposing
C. MapReduce
D. None of above

8. Which of the following method causes call returns only when the job gets finished, and it returns with its success or failure status which can be used to determine that further steps are to be run or not?

A. Waitforfinished()
B. waitForCompletion()
C. Both
D. None of the above

9. Which of the following specifies the environment variables that affect the JDK used by Hadoop Daemon (bin/hadoop).

A. core-site.xml
B. hadoop-env.sh
C. hdfs-site.xml
D. mapred-site.xml

10. Which of the followings are important configuration files which is required for runtime environment settings of a Hadoop cluster that also informs Hadoop daemons where the NAMENODE runs in the cluster.

A. core-site.xml
B. hadoop-env.sh
C. hdfs-site.xml
D. mapred-site.xml

11. _____ is responsible for scheduling tasks, monitoring them, and re-executes the failed task.

A. Hadoop MapReduce
B. Yarn
C. Hive
D. Pig

12. YARN stands for _____

A. Yet Another Router Negotiator
B. Yet Another Resource Network
C. Yet Another Resource Negotiator
D. None of above

13. In Hadoop, the process by which the intermediate output from mappers is transferred to the reducer is called _____.

A. Shuffling

B. Sorting
C. Both
D. None of above

14. Which of the following's tasks are the first phase of processing that processes each input record and generates an intermediate key-value pair?
A. Reducer task
B. Mapper Task
C. Compress Task
D. None of above

15. Which of the following phases occur simultaneously?
A. Reduce and Sort
B. Map and Reduce
C. Shuffle and Sort
D. All of the mentioned

Answers for Self Assessment

1.	A	2.	B	3.	D	4.	A	5.	D
6.	C	7.	B	8.	B	9.	B	10.	A
11	B	12.	C	13.	A	14.	B	15.	C

Review Questions

1. Explain all unit testing techniques.
2. Explain three core classes of MRUNIT.
3. Explain Developing and testing MapReduce jobs with MRUnit
4. Diagrammatically explain shuffle and sort concepts
5. Explain three kinds of failure in MapReduce.

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
- of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/
7. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm

Unit 11: Hadoop Ecosystem

CONTENTS

Objectives

- explore the concepts of HIVE.
- understand architecture of HIVE.
- explore concepts and architecture of Apache pig
- understand Pig-Latin data types, applications and features of Pig
- learn operators in Apache pig.
- learn services offered by HIVE
- Learn fundamentals of Hbase
- Explore concepts of ZooKeeper
- understand IBM InfoSphere Streams
- learn a new paradigm for information processing, learn powerful, real-time analytic processing made simple
- Explore Enterprise integration and concepts of scale-out architecture.

Introduction

Apache Hive is an opensource data warehousing solution built on the Hadoop platform. Hive is a database that may be used to analyse and query huge datasets contained in Hadoop files. Hive may be used to process both structured and semi-structured data.On the BigData landscape, Apache Hive is one of the most used datawarehouse components. Its interface is mostly used to supplement the Hadoop file system.Hive was created by Facebook and is currently maintained by the Apache Software Foundation as Apache Hive.Netflix and Amazon are among the companies that utilise and develop it.Hive is a Hadoop data warehouse architecture solution that allows you to handle structured data. It resides on top of Hadoop to summaries Big Data and facilitate searching and analysis.It resides on top of Hadoop to summaries Big Data and facilitate searching and analysis.Initially created by Facebook, Hive was eventually taken up by the Apache Software Foundation and maintained as an open-source project under the name Apache Hive. It is utilized by a variety of businesses. For example, Amazon utilizes it in Amazon Elastic MapReduce.

Hive is not

- A language for real-time queries
- and row-level updates

11.1 Features of HIVE

The schema is stored in a database, and the processed data is saved to HDFS as shown in Figure 1. It was created with OLAP in mind. It has a querying language called HiveQL or HQL that is similar to SQL. It's easy to use, quick, scalable, and extendable.

Figure 1 Features of HIVE

11.2 Architecture of HIVE

Bigdata developers must always have a basic understanding of how all of the components operate together. That's where we learn how the components operate and how to write code that adheres to them without causing performance issues. Hive's general architecture is depicted in this diagram. You can simply recall how Hive works if you look at this architectural diagramFigure 2.

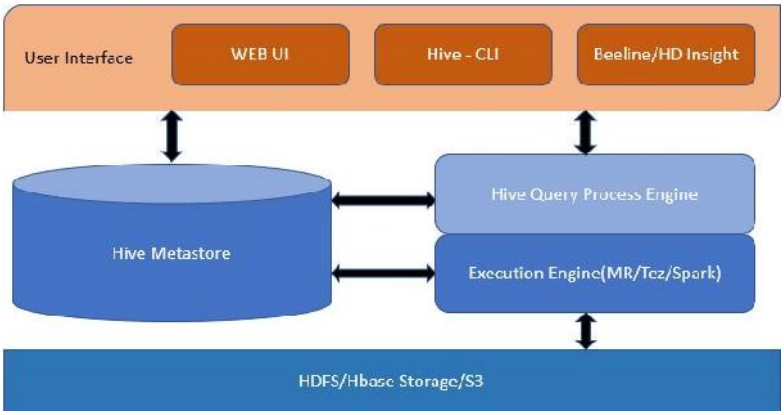


Figure 2 Architecture of HIVE

- ***User Interface:***This is where the end user interacts with Hive in order for the data to be processed. We offer various methods to interface with Hive, including the Web UI and the Hive CLI, which is included with the Hive package

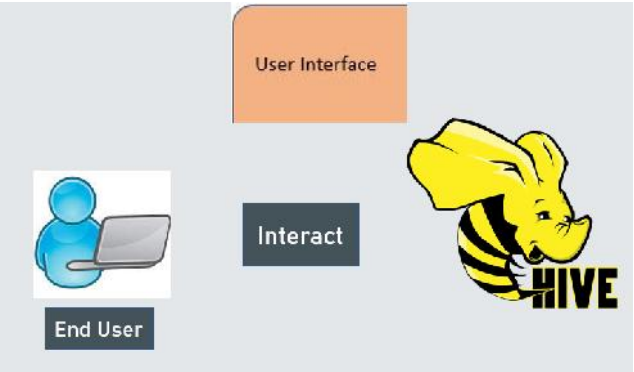


Figure 3 User Interface

We may also use Thrift Client, JDBC Client, and ODBC Client. Hive also offers services such as Hive CLI, Beeline, and others.

- ***Hive Query process engine***: The query entered via the user interface is parsed by the Hive compiler. It uses information contained in the metastore to verify for semantic and syntactic accuracy. Finally, it generates an execution plan in the form of a DAG (Directed Acyclic Graph), with each stage representing a mapreduce task as shown in Figure 4.

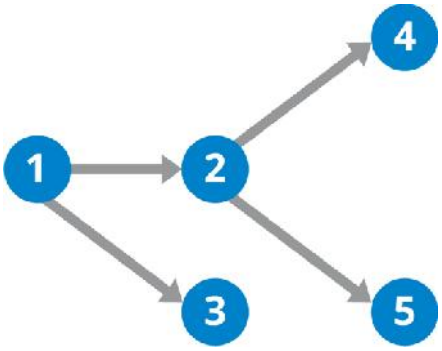


Figure 4 Directed Acyclic Graph

- ***Execution Engine***: Execution Engine is where the actual processing of the data will start. After compiler checking the syntax, performs the optimizations of the execution. Finally, this execution plan will be given to Execution Engine.We have several execution engines that can be used with Hive. MapReduce is one of the execution engines which slower compared other engines. We can change to this execution engine to Tez or Spark. To change the execution engine we can use the below command:

```
set hive.execution.engine=spark;

set hive.execution.engine=tez;

set hive.execution.engine=mr;
```

- ***Metastore:***

  Metastore is the central repository where the metadata about tables will be stored. This metadata includes database names. table names, column details along with data types of columns, and table partition details. It also stores the details about Serialization and Deserialization details of the files stored in underlying storage system. In generalmetastore is relational database. Metastore provides thrift server to interact with it.Metastore can be used in two modes.

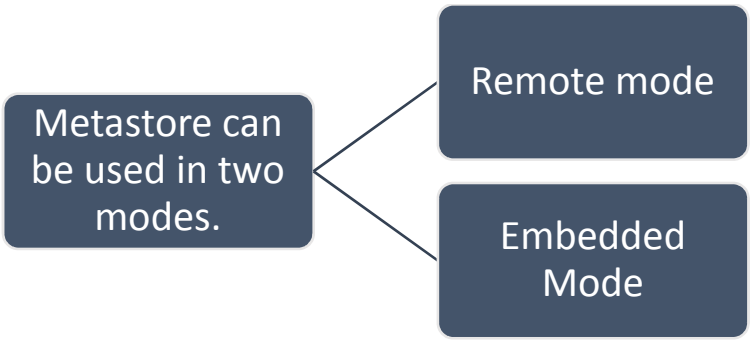


Figure 5 Two modes of metastore

- **Remote mode**: In this mode meta-store is a Thrift Service which can be used in case non-Java applications.
- **Embedded Mode**: In this case client can directly interact with meta-store using JDBC.

HDFS/Hbase Storage:

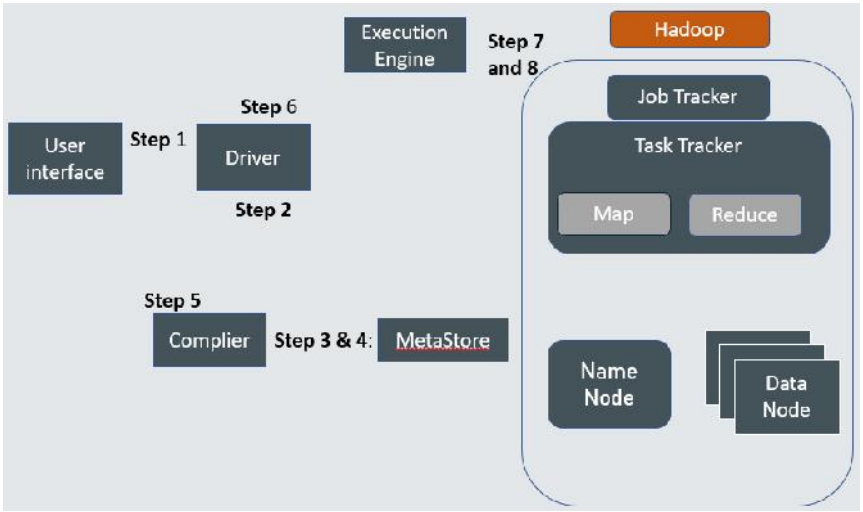
Hive is unable to store data directly. Hive can only analyses data and enter it into tables; the data itself is kept on storage systems such as HDFS, HBase, or S3. Hive will create tables that point to the location of the data in any of the above storage systems, and the data will be accessed from there.

- UI (User Interface): The user interface is for users to submit queries and other operations to the system.
- Driver: This is the component that receives the requests. This component supports the concept of session handles and provides JDBC/ODBC-style execute and fetch APIs.
- Compiler: The component that parses the query does semantic analysis on the various query blocks and query expressions, and then creates an execution plan using information from the Metastore for table and partition metadata.
- Metastore: The component that holds all of the structural information for the warehouse's different tables and partitions, including column and column type information, serializers and de-serializers for reading and writing data, and the HDFS files where the data is kept.
- Execution Engine: The component responsible for carrying out the compiler's execution plan. The strategy is organized as a DAG of phases. The execution engine coordinates the dependencies between these various plan stages and performs them on the relevant system components.
- This metadata is used to type check the expressions in the query tree as well as to prune partitions based on query predicates. The plan generated by the compiler is a DAG of stages with each stage being either a map/reduce job, a metadata operation or an operation on HDFS. For map/reduce stages, the plan contains map operator trees (operator trees that are executed on the mappers) and a reduce operator tree (for operations that need reducers)., a metadata operation or an operation on HDFS. For map/reduce stages, the plan contains map operator trees (operator trees that are executed on the mappers) and a reduce operator tree (for operations that need reducers).

11.3 Working of HIVE

We are aware of all the components of hive and their functionalities. So now let's see the working of hive

- **Step** 1: The UI calls the execute interface to the Driver.
- **Step 2**: The Driver creates a session handle for the query and sends the query to the compiler to generate an execution plan.
- **Step 3 & 4**: The compiler gets the necessary metadata from the Metastore.
- **Step 5**: This metadata is used to type check the expressions in the query tree as well as to prune partitions based on query predicates. The plan generated by the compiler is a DAG of stages with each stage being either a map/reduce job, a metadata operation or an operation on HDFS. For map/reduce stages, the plan contains map operator trees (operator trees that are executed on the mappers) and a reduce operator tree (for operations that need reducers).
- **Step 6**: The execution engine submits these stages to appropriate components (steps 6, 6.1, 6.2 and 6.3). In each task (mapper/reducer) the deserializers associated with the table or intermediate outputs is used to read the rows from HDFS files and these are passed through the associated operator tree. Once the output is generated, it is written to a temporary HDFS file though the serializers (this happens in the mapper in case the operation does not need a reduce). The temporary files are used to provide data to subsequent map/reduce stages of the plan. For DML operations the final temporary file is moved to the table's location.
- **Step 7 & 8 & 9**: For queries, the contents of the temporary file are read by the execution engine directly from HDFS as part of the fetch call from the Driver.Once the output is generated, it is written to a temporary HDFS file though the serializers (this happens in the mapper in case the operation does not need a reduce). The temporary files are used to provide data to subsequent map/reduce stages of the plan. For DML operations the final temporary file is moved to the table's location.



Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

- What is HIVE in Hadoop?

With Hadoop Hive, you can just go ahead and submit SQL queries and perform MapReduce jobs. So, if you are comfortable with SQL, then Hive is the right tool for you as you will be able to work on MapReduce tasks efficiently. Similar to Pig, Hive has its own language, called HiveQL (HQL). It is similar to SQL. HQL translates SQL-like queries into MapReduce jobs, like what Pig Latin does. The best part is that you don't need to learn Java to work with Hadoop Hive. Hadoop Hive runs on our system and converts SQL queries into a set of jobs for execution on a Hadoop cluster. Basically, Hadoop Hive classifies data into tables providing a method for attaching the structure to data stores in HDFS.

11.4 Introduction to Apache Pig

Hadoop use Map Reduce to analyse and process large amounts of data. A Java application called Map Reduce is used to reduce the size of a file.
Developers, on the other hand, find it difficult to build and maintain these long Java scripts. Develo pers may use Apache Pig to swiftly examine and
handle big data sets without having to write complicated Java code.Apache Pig developed by Yahoo researchers executes Map Reduce jobs on extensive datasets and provides an easy interface for developers to process the data efficiently. Apache Pig emerged as a boon for those who do not understand Java programming. Today, Apache Pig has become very popular among developers as it offers flexibility, reduces code complexity, and requires less effort.

11.5 Why the name PIG?

The Apache Pig programming language, like pigs who eat anything, is designed to operate with any type of data. That's how Pig got his name!

Figure 6 MapReduce vs Apache Pig

11.6 Architecture of PIG

The architecture of PIG includes following: -

Figure 7 Architecture of Pig

In Pig, there is a language we use to analyze data in Hadoop. That is what we call Pig Latin. Also, it is a high-level data processing language that offers a rich set of data types and operators to perform several operations on the data.Moreover, in order to perform a particular task, programmers need to write a Pig script using the Pig Latin language and execute them using any of the execution mechanisms (Grunt Shell, UDFs, Embedded) using Pig. To produce the desired output, these scripts will go through a series of transformations applied by the Pig Framework, after execution.

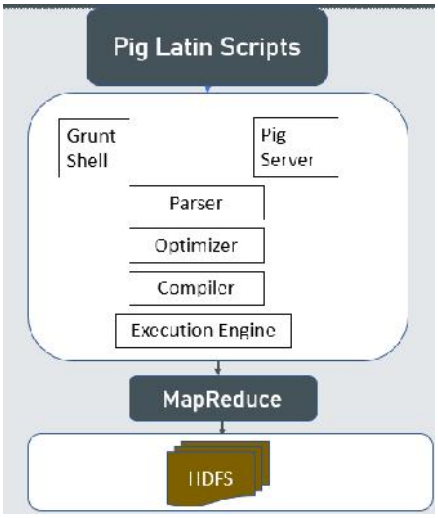


Figure 8 Pig Latin Scripts

Further, Pig converts these scripts into a series of MapReduce jobs internally. Therefore, it makes the programmer's job easy. Here, is the architecture of Apache Pig.

a. **Parser**: At first, all the Pig Scripts are handled by the Parser. Parser basically checks the syntax of the script, does type checking, and other miscellaneous checks. Afterwards, Parser's output will be a DAG (directed acyclic graph) that represents the Pig Latin statements as well as logical operators.The logical operators of the script are represented as the nodes and the data flows are represented as edges in DAG (the logical plan)

Figure 9 Parser

b. **Optimizer**: Afterwards, the logical plan (DAG) is passed to the logical optimizer. It carries out the logical optimizations further such as projection and push down.

c. **Compiler**: Afterwards, the logical plan (DAG) is passed to the logical optimizer. It carries out the logical optimizations further such as projection and push down.

d. **Execution Engine**: Eventually, all the MapReduce jobs are submitted to Hadoop in a sorted order. Ultimately, it produces the desired results while these MapReduce jobs are executed on Hadoop.

Pig Latin data model

Pig Latin data model is fully nested. Also, it allows complex non-atomic data types like map and tuple. Let's discuss this data model in detail:

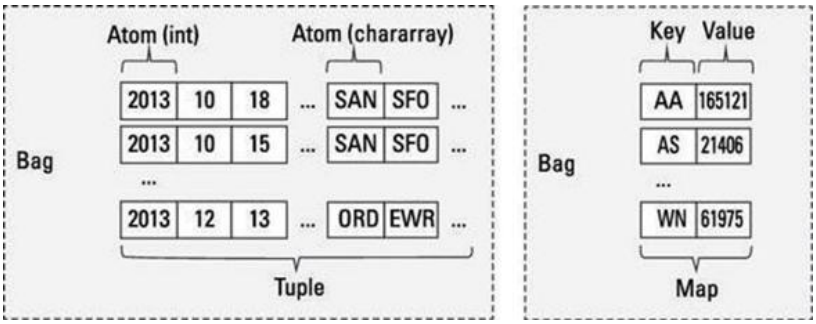


Figure 10 Atom

a. **Atom**: Atom is defined as any single value in Pig Latin, irrespective of their data. Basically, we can use it as string and number and store it as the string. Atomic values of Pig are int, long, float, double, char array, and byte array. Moreover, a field is a piece of data or a simple atomic value in Pig. For Example − 'Shubham' or '25'

b. **Bag:**An unordered set of tuples is what we call Bag. To be more specific, a Bag is a collection of tuples (non-unique). Moreover, each tuple can have any number of fields (flexible schema). Generally, we represent a bag by '{}'.It is similar to a table in RDBMS. It is not necessary that every tuple contain the same number of fields in the same position(column) have the same type.

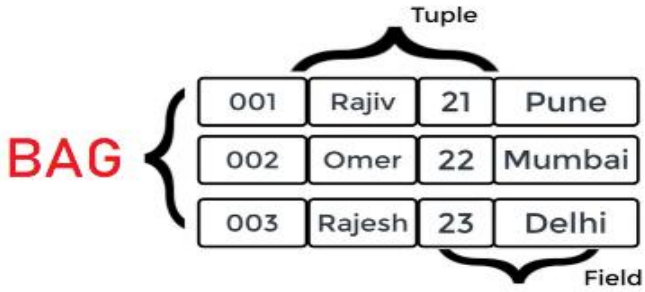


Figure 11 Bag

11.7 Pig Latin Data Type

We will understand what are the data types are now. Whatever we write will be given in the data. Type so pig contains basic data types like int, float, double. Now here we don't say string we say chararray and its string format and then you have got a byte array.

Table 1 Pig-Latin Data Types

Data Type	Description	Example
Int	Represent a signed 32-bit integer	8
long	Represent a signed 64-bit integer	5L
float	Represent a signed 32-bit integer	5.5L
double	Represent a signed 64-bit integer	10.5L
Chararray	Represents a character array(string) in Unicode UTF-8 format	'Online Classes'
Bytearray	Represents a Byte array(blob)	
Boolean	Represents a Boolean values	True/false
Datetime	Represents a Date-time.	1970-01-01T00:00:00.000+00:00
BigInteger	Represents a java Biginteger	60708090709
Bigdecimal	Represents a Bigdecimal	185.98376256272893883

1. The byte array is your blob file. Remember one thing if you do not provide any data right while making your application. It will be by default considered as a byte array schema. You have got Boolean which consist of true and false.
2. Data/time now like the whole timestamp like thing then you have a big integer and big decimal for taking the value bigger than double and floats. Now you have also some of complex types.

Table 2 Complex Types

Data Type	Description	Example
Tuple	A tuple is an ordered set of fields	(raja,30)
Bag	A bag is a collection of tuples	{(raju,30),(Mohammad,45)}
Map	A Map is a set of key-value pairs	['name'#'Raju','age'#30]

3. Tuple is basically a single row which will be in a round brackets. In the bag you have got curly braces. Curly braces contain lots of other tuples inside it. Tuple is represented using round bracket and bag is represented using a curly bracket
4. A map is basically is a combination of key-value pair which makes up of your key-values pairs which is separated by a hash tag. A hash tag is a separator between y our map key value pairs.
5. Values for all the above data types can be NULL. Apache Pig treats null values in a similar way as SQL does. A null can be an unknown value or a non-existent value. It is used as a

placeholder for optional values. These nulls can occur naturally or can be the result of an operation.

Pig Latin – Arithmetic Operators

Bincond:

- Evaluates the Boolean operators. It has three operands as shown below.
- variable **x** = (expression) ? **value1** *if true* : **value2** *if false*.
- b = (a == 1)? 20: 30; if a = 1 the value of b is 20. if a!=1 the value of b is 30.

Case − The case operator is equivalent to nested bincond operator.

CASE f2 % 2 WHEN 0 THEN 'even' WHEN 1 THEN 'odd' END

Pig Latin – Comparison Operators

Table 3 Pig Latin-Comparison Operators

==Equal
!=Not Equal
>Greater than
<Less than
>=Greater than or equal to
<=Less than or equal to
Matches(Pattern matching)

Pig Latin – Construction Operators

Table 4 Pig Latin - Construction Operators

Operator- Description
() - Tuple constructor operator
{} - Bag constructor operator
[] - Map constructor operator

Pig Latin – Relational Operators

Table 5 Pig Latin - Relational Operators

Operator- Description
Loading and Storing
LOAD
STORE
Filtering
FILTER
DISTINCT
FOREACH, GENERATE
STREAM
Grouping and Joining
JOIN
COGROUP
GROUP
CROSS
Sorting
ORDER
LIMIT
Combining and Splitting
UNION
SPLIT
Diagnostic Operators
DUMP
DESCRIBE
EXPLAIN
ILLUSTRATE

11.8 Applications of Apache PIG

For exploring large datasets Pig Scripting is used. Provides the supports across large data-sets for Ad-hoc queries. In the prototyping of large data-sets processing algorithms. Required to process the

time sensitive data loads.For collecting large amounts of datasets in form of search logs and web crawls. Used where the analytical insights are needed using the sampling.

Apache Pig comes with the following features −

- **Rich set of operators** − It provides many operators to perform operations like join, sort, filer, etc. **Ease of programming** − Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL.
- **Optimization opportunities** − The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language. **Extensibility** − Using the existing operators, users can develop their own functions to read, process, and write data.
- **UDF's** − Pig provides the facility to create **User-defined Functions** in other programming languages such as Java and invoke or embed them in Pig Scripts. **Handles all kinds of data** − Apache Pig analyzes all kinds of data, both structured as well as unstructured. It stores the results in HDFS.

Features of Apache PIG

a. Rich set of operators
b. Ease of programming
c. Optimization opportunities
d. Extensibility
e. UDF's
f. Handles all kinds of data

11.9 Operators in Apache PIG

It is a high-level procedural language for querying large data sets using Hadoop and the Map Reduce Platform. It is a Java package, where the scripts can be executed from any language implementation running on the JVM. It simplifies the use of Hadoop by allowing SQL-like queries to a distributed dataset. It backs many relational features like Join, Group and Aggregate.It does have many features common with ETL tools. The Apache Pig has two execution modes as shown inFigure 12.

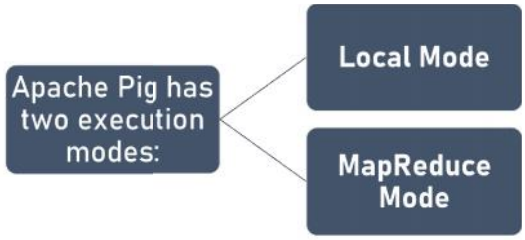


Figure 12 Two execution modes in Apache Pig

Local Mode

The source data for 'Local Mode' would be taken from your computer's local directory. The 'pig –x local' command can be used to specify the MapReduce mode.

MapReduce Mode

You'll need access to a Hadoop cluster and an HDFS installation to run Pig in MapReduce mode. The 'pig' command may be used to specify the MapReduce mode.

Apache Pig Operators

The Apache Pig Operators are a high-level procedural language that may be used to query big data sets utilizing Hadoop and the Map Reduce Platform. A Pig Latin statement is a type of operator that accepts one relation as input and outputs another. These operators are the most important tools Pig Latin gives for working with data. Sorting, grouping, joining, projecting, and filtering are all options for transforming it.

Classification of Apache Pig operators

There are two types of Apache Pig operators: relational and diagnostic.

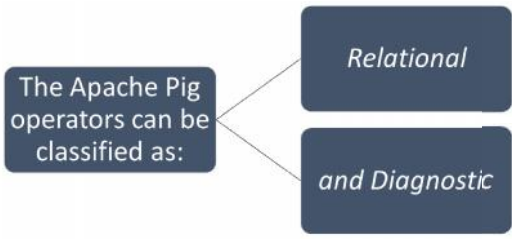


Figure 13 Classification of Apache Pig Operators

Relational Operators

Pig Latin's main tools for manipulating data are relational operators. Sorting, grouping, merging, projecting, and filtering the data are all options. The basic relational operators are:-

LOAD

The LOAD operator is used to insert data into a Pig relation from a file system or HDFS storage.

FOREACH

Based on data columns, this operator creates data transformations. It's used to make changes to a relation's fields. To interact with data columns, use the FOREACH-GENERATE method.

FILTER

This operator uses a condition to pick tuples from a relation.

JOIN

The JOIN operator is used to accomplish an inner, equijoin join of two or more relations based on field values that are the same. An inner join is always performed by the JOIN operator. Null keys are ignored by inner joins, therefore filtering them out before the join makes logical.

ORDER BY

Order By is a feature that allows you to sort a relation by one or more fields. Using the ASC and DESC keywords, you may sort in ascending or descending order.

DISTINCT

In a relation, distinct eliminates duplicate tuples. Consider the following input file, which contains amr,crap,8 and amr,myblog,10 twice. Duplicate items are deleted when distinct is applied to the data in this file.

STORE

The term "store" refers to the process of saving results to a file system.

GROUP

The GROUP operator joins tuples that have the same group key (key field). If the group key includes more than one field, the key field will be a tuple; otherwise, it will be the same type as the group key. A GROUP operation produces a relation that has one tuple per group.

CROSS

To compute the cross product (Cartesian product) of two or more relations, use the CROSS operator.

LIMIT

To restrict the number of output tuples, use the LIMIT operator. The output will include all tuples in the relation if the provided number of output tuples is equal to or more than the number of tuples in the relation.

SPLIT

The SPLIT operator divides a relation's contents into two or more relations depending on an expression. In accordance with the requirements indicated in the statement.

Diagnostic Operators

DUMP

The DUMP operator executes Pig Latin commands and displays the results on the screen.

DESCRIBE

To study the schema of a specific relation, use the DESCRIBE operator. When troubleshooting a script, the DESCRIBE operator is very useful.

ILLUSTRATE

The ILLUSTRATE operator is used to see how data is changed using Pig Latin statements. When it comes to debugging a script, the ILLUSTRATE command is your best friend. This command alone could be enough to convince you to use Pig instead of something else.

EXPLAIN

The logical and physical planes are printed by the EXPLAIN operator.

11.10 Introduction to HIVEQL

Apache Hive is an open-source data warehousing platform that may be used to execute distributed processing and analysis. It was created by Facebook to make creating the Java MapReduce application easier.The Hive Query language, which is a declarative language comparable to SQL, is used by Apache Hive. Hive is a programme that converts hive searches into MapReduce programmes.It allows developers to process and analyse structured and semi-structured data by removing the need for sophisticated Java Maps. Using hive queries, you can cut down on the number of programmes you run.The hive queries are simple to create for someone who is familiar with SQL commands.The hive queries are simple to create for someone who is familiar with SQL commands.

- Hive makes it simple to conduct procedures such as
    - Analysis of huge datasets, Ad-hoc queries, Data encapsulation

The major components of Apache Hive are:

- **Hive Client, Hive Services, Processing and Resource Management, Distributed Storage**

For running queries on the Hive, Hive supports applications written in any language, including Python, Java, C++, Ruby, and others, using JDBC, ODBC, and Thrift drivers. As a result, writing a hive client application in any language of one's choice is simple.

Services Offered by HIVE

Hive provides several services, including as the

1. Beeline,
2. Hive Server 2
3. Hive Driver
4. Hive Compiler
5. Optimize, Execution Engine,

6. Metastore 1. **Remote 2. Embedded**
7. Hcatalog, WebHCat

Beeline

Beeline is **a command line interface of hive server2 a new launched product of hive**. ... Recently, the Hive community introduced HiveServer2 which is an enhanced Hive server designed for multi-client concurrency and improved authentication that also provides better support for clients connecting through JDBC and ODBC

Hive Server2

HiveServer2 is HiveServer1's successor. Clients can use HiveServer2 to run queries against Hive. It enables numerous clients to send Hive requests and retrieve the results. Its primary purpose is to provide the greatest possible support for open API clients such as JDBC and ODBC.The Thrift-based Hive service is at the heart of HS2 and is in charge of handling Hive queries (e.g., from Beeline). Thrift is a cross-platform RPC framework for creating services. Server, Transport, Protocol, and Processor are the four levels that make up its stack.

Hive Driver

The Hive driver accepts the HiveQL statements entered into the command shell by the user. It generates the query's session handles and sends it to the compiler. Hive Driver is a Java Script driver for connection to **Apache Hive** via **Thrift API**. This driver can connect with SASL authentication mechanisms (such as LDAP, PLAIN, Kerberos) using both HTTP and TCP transport.

Hive Compiler

The query is parsed by the Hive compiler. It uses the metadata stored in the metastore to do semantic analysis and type-checking on the various query blocks and query expressions, and then generates an execution plan.Hive Compiler is a tool that allows you to compile data in Hive. The DAG (Directed Acyclic Graph) is the execution plan generated by the compiler, with each step consisting of a map/reduce job, an HDFS action, and a metadata operation.Optimizer separates the job and performs transformation operations on the execution plan to increase efficiency and scalability.
Compiler communicating with Driver with the proposed plan to execute the query.

Optimize

Optimizer performs the transformation operations on the execution plan and splits the task to improve efficiency and scalability.optimize. bucketmapjoin=true. This setting hints to Hive to do bucket level join during the map stage join. It also reduces the scan cycles to find a particular key because bucketing ensures that the key is present in a specific bucket. It optimises and transforms an execution plan in order to produce an optimal Directed Acyclic Graph (DAG). To improve performance and scalability, transformations such as transforming a pipeline of joins to a single join and job separation, such as putting a transformation on data before a reduce operation, are used.

Optimization Techniques:

Let's go over each of the Hive optimization strategies for Hive Performance Tuning one by one:

a. **Tez-Execution Engine in HIVE:**Hive Optimization Techniques, to increase the Hive performance of our hive query by using our execution engine as Tez. On defining Tez, it is a new application framework built on **Hadoop Yarn**. That executes complex-directed acyclic graphs of general data processing tasks. However, we can consider it to be a much more flexible and powerful successor to the map-reduce framework.In addition, to write native YARN applications on Hadoop that bridges the spectrum of interactive and batch workloads Tez offers an API framework to developers. To be more specific, to work with petabytes of data over thousands of nodes it allows those data access applications.

b. **Usage of Suitable File Format in Hive**: Hive Optimization Techniques, if we use appropriate file format on the basis of data. It will drastically increase our query performance. Basically, for increasing your query performance ORC file format is best suitable. Here, ORC refers to Optimized Row Columnar. That implies we can store data in

an optimized way than the other file formats. To be more specific, ORC reduces the size of the original data up to 75%. Hence, data processing speed also increases. On comparing to Text, Sequence and RC file formats, ORC shows better performance.Basically, it contains rows data in groups. Such as Stripes along with a file footer. Therefore, we can say when Hive is processing the data ORC format improves the performance.To be more specific, ORC reduces the size of the original data up to 75%. Hence, data processing speed also increases. On comparing to Text, Sequence and RC file formats, ORC shows better performance. Basically, it contains rows data in groups. Such as Stripes along with a file footer. Therefore, we can say when Hive is processing the data ORC format improves the performance.

c. **Hive Partitioning**: Hive Optimization Techniques, Hive reads all the data in the directory Without partitioning. Further, it applies the query filters on it. Since all data has to be read this is a slow as well as expensive. Also, users need to filter the data on specific column values frequently. Although, users need to understand the domain of the data on which they are doing analysis, to apply the partitioning in the Hive.Basically, by Partitioning all the entries for the various columns of the dataset are segregated and stored in their respective partition. Hence, While we write the query to fetch the values from the table, only the required partitions of the table are queried. Thus, it reduces the time taken by the query to yield the result.

d. **Bucketing in Hive**: Hive Optimization Techniques, let's suppose a scenario. At times, there is a huge dataset available. However, after partitioning on a particular field or fields, the partitioned file size doesn't match with the actual expectation and remains huge.Still, we want to manage the partition results into different parts. Thus, to solve this issue of partitioning, Hive offers Bucketing concept. Basically, that allows the user to divide table data sets into more manageable parts.Hence, to maintain parts that are more manageable we can use Bucketing. Through it, the user can set the size of the manageable parts or Buckets too.

e. **Vectorization In Hive**: Hive Optimization Techniques, to improve the performance of operations we use Vectorized query execution. Here operations refer to scans, aggregations, filters, and joins. It happens by performing them in batches of 1024 rows at once instead of single row each time.However, this feature is introduced in Hive 0.13. It significantly improves query execution time, and is easily enabled with two parameters settings:

set hive.vectorized.execution = true

set hive.vectorized.execution.enabled = true

f. **Cost-Based Optimization in Hive (CBO):**Hive Optimization Techniques, before submitting for final execution Hive optimizes each Query's logical and physical execution plan. Although, until now these optimizations are not based on the cost of the query.However, CBO, performs, further optimizations based on query cost in a recent addition to Hive. That results in potentially different decisions: how to order joins, which type of join to perform, the degree of parallelism and others.To use CBO, set the following parameters at the beginning of your query:

```
set hive.cbo.enable=true;
set hive.compute.query.using.stats=true;
set hive.stats.fetch.column.stats=true;
set hive.stats.fetch.partition.stats=true;
```

Then, prepare the data for CBO by running Hive's "analyze" command to collect various statistics on the tables for which we want to use CBO.

g. **Hive Indexing:**Hive Optimization Techniques, one of the best ways is Indexing. To increase your query performance indexing will definitely help. Basically, for the original table use of indexing will create a separate called index table which acts as a reference. As we know, there are many numbers of rows and columns, in a **Hive table**. Basically, it will take a large amount of time if we want to perform queries only on some columns without indexing. Because queries will be executed on all the columns present in the table. Moreover, there is no need for the query to scan all the rows in the table while we perform a query on a table that has an index, it turned out as the major advantage of using indexing. Further, it checks the index first and then goes to the particular column and performs the operation.Hence, maintaining indexes will be easier for Hive query to look into the indexes first and then perform the needed operations within less amount of time. Well, time is the only factor that everyone focuses on, eventually.

h. **Execution Engine:** The execution engine uses Hadoop to execute the execution plan created by the compiler in order of their dependencies following the compilation and optimization processes.

i. **MetaStore:** It also holds serializer and deserializer metadata, which is essential for read/write operations, as well as HDFS files where data is kept. In most cases, this metastore is a relational database.For searching and altering Hive metadata, Metastore provides a Thrift interface.It also holds serializer and deserializer metadata, which is essential for read/write operations, as well as HDFS files where data is kept. In most cases, this metastore is a relational database.For searching and altering Hive metadata, Metastore provides a Thrift interface.Metastore can be configured in one of two ways:

Remote: Metastore is a Thrift service in remote mode, which is suitable for non-Java applications.

Embedded: In embedded mode, the client can use JDBC to interface directly with the metastore.

j. **HCatalog**: Hadoop's table and storage management layer is HCatalog. It allows users to read and write data on the grid using various data processing tools such as Pig, MapReduce, and others.It is based on Hive metastore and exposes Hive metastore's tabular data to other data processing tools.HCatalog is a Hadoop table and storage management layer that allows users to read and write data on the grid more simply using various data processing tools such as Pig and MapReduce. Users can have a relational view of data in the Hadoop distributed file system (HDFS) thanks to HCatalog's table abstraction, which means they don't have to worry about where or how their data is stored — RCFile format, text files, SequenceFiles, or ORC files.HCatalog supports reading and writing files in any format for which a SerDe (serializer-deserializer) can be written. By default, HCatalog supports RCFile, CSV, JSON, and SequenceFile, and ORC file formats. To use a custom format, you must provide the InputFormat, OutputFormat, and SerDe.

k. **WebHCat**: For HCatalog, WebHCat is the REST API. It's a Hive metadata operations HTTP interface. It lets users perform Hadoop MapReduce (or YARN), Pig, and Hive jobs.Developers use HTTP requests to access Hadoop MapReduce (or YARN), Pig, Hive, and HCatalog DDL from within applications, as demonstrated in the diagram below. HDFS stores the data and code utilised by this API. When HCatalog DDL commands are requested, they are immediately executed. WebHCat (Templeton) servers queue

MapReduce, Pig, and Hive jobs, which may be monitored for progress or cancelled as needed. Pig, Hive, and MapReduce results are stored in HDFS, and developers select where they should be stored.

11.11 HIVEQL

Hive Query Language (HiveQL) is a query language for Hive that allows you to process and analyse structured data in a Metastore.A table's data is retrieved using the SELECT command. The WHERE clause functions in the same way as a condition. It applies the condition to the data and returns a finite result. The built-in operators and functions provide an expression that meets the criteria.The SELECT query's syntax is as follows:

SELECT [ALL | DISTINCT] select_expr, select_expr, ...

FROM table_reference

[WHERE where_condition]

[GROUP BY col_list]

[HAVING having_condition]

[CLUSTER BY col_list | [DISTRIBUTE BY col_list] [SORT BY col_list]]

[LIMIT number];

ORDER BY clause in a SELECT statement:

The ORDER BY clause is used to get information about a single column and sort the results in ascending or descending order.

Syntax:

SELECT [ALL | DISTINCT] select_expr, select_expr, ...

FROM table_reference

[WHERE where_condition]

[GROUP BY col_list]

[HAVING having_condition]

[ORDER BY col_list]]

[LIMIT number];

GROUP BY clause in a SELECT statement

The GROUP BY clause is used to group all of the records in a result set by a collection column. It's used to look up a set of records.

Syntax:

SELECT [ALL | DISTINCT] select_expr, select_expr, ...

FROM table_reference

[WHERE where_condition]

[GROUP BY col_list]

[HAVING having_condition]

[ORDER BY col_list]]

[LIMIT number];

JOIN clause

JOIN is a clause that is used for combining specific fields from two tables by using values common to each one. It is used to combine records from two or more tables in the database.

Syntax:

join_table:

table_reference JOIN table_factor [join_condition]

| table_reference {LEFT|RIGHT|FULL} [OUTER] JOIN table_reference

join_condition

| table_reference LEFT SEMI JOIN table_referencejoin_condition

| table_reference CROSS JOIN table_reference [join_condition]

HBase

Since 1970, RDBMS has been the go-to option for data storage and maintenance issues. Companies discovered the value of processing big data after the introduction of big data and began to use technologies such as Hadoop.Hadoop stores massive data in a distributed file system and processes it using MapReduce. Hadoop excels at storing and processing large amounts of data in a variety of formats, including random, semi-structured, and unstructured data.Tables in HBase are divided into regions and served by region servers. Column families divide regions vertically into "Stores." In HDFS, stores are saved as files. HBase's architecture is depicted below.

Architecture of HBase

HBase provides low-latency random reads and writes on top of HDFS. In HBase, tables are dynamically distributed by the system whenever they become too large to handle (Auto Sharding). The simplest and foundational unit of horizontal scalability in HBase is a Region. A continuous, sorted set of rows that are stored together is referred to as a region (subset of table data). HBase architecture has a single HBase master node (HMaster) and several slaves i.e. region servers. Each region server (slave) serves a set of regions, and a region can be served only by a single region server. Whenever a client sends a write request, HMaster receives the request and forwards it to the corresponding region server.
HBase can be run in a multiple master setup, wherein there is only single active master at a time. HBase tables are partitioned into multiple regions with every region storing multiple table's rows.

MasterServer

- This process is accomplished with the help of Apache ZooKeeper, which assigns regions to region servers.Handles region load balancing across area servers. It moves the regions to less crowded servers after unloading the congested servers.By negotiating load balancing, it keeps the cluster in good shape.Is in charge of schema modifications and other metadata actions like table and column family formation.

Regions

- Tables are broken up and distributed across area servers to form regions.
- There are regions on the region servers that -

  Handle data-related actions and communicate with the client.
  For all the areas beneath it, handle read and write requests.
  Follow the region size thresholds to determine the region's size.

When we take a closer look at the region server, we can see that it has the following regions and stores:

Memory and HFiles are both stored in the store. Memstore works similarly to cache memory. Everything that is entered into HBase is initially saved here. The data is then transported and saved as blocks in Hfiles, and the memstore is flushed.

HBase Data Models

The HBase data model holds semi-structured data with a variety of data types, as well as different column and field sizes. The HBase data model's structure makes data segmentation and distribution throughout the cluster a breeze. Row key, column family, table name, timestamp, and other logical components make up the HBase data model. In HBase tables, the Row Key is used to uniquely identify the rows. HBase's column families are static, while the columns themselves are dynamic.

- ***HBase Tables***: Logical collection of rows stored in individual partitions known as regions.
- ***HBase Row***: Instance of data in a table.
- ***Row Key:*** Every entry in an HBase table is identified and indexed by a Row Key.
- ***Columns:*** For every Row Key an unlimited number of attributes can be stored.
- ***Column Family***: Data in rows is grouped together as column families and all columns are stored together in a low-level storage file known as HFile.

Components of Apache HBase Architecture

HMaster is the HBase implementation of Master Server. It is a process in which regions and DDL (new, delete table) operations are assigned to a region server. It keeps track of all Region Server instances in the cluster.Master runs numerous background threads in a distributed system. HMaster includes a lot of functions, such as load balancing, failover, and so on.

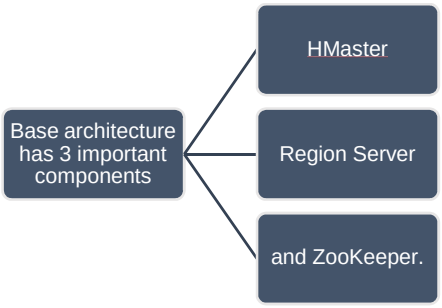


Figure 14 Apache HBase Architecture

Region Server

HBase Tables are separated into Regions horizontally by row key range. Regions are the fundamental building blocks of an HBase cluster, consisting of a distribution of tables and Column families. The Region Server runs on an HDFS DataNode in the Hadoop cluster. Region Server regions are responsible for a variety of tasks, including handling, administering, and performing HBase operations on that set of regions. A region's default size is 256 MB.

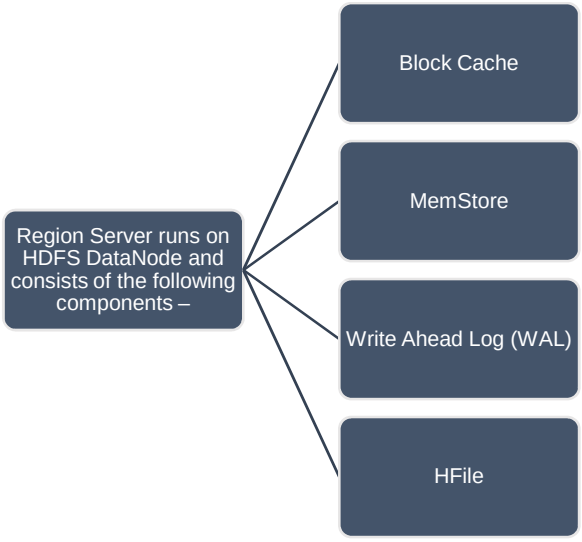


Figure 15 Components of Region Server

- Block Cache – The read cache is located here. The read cache stores the most frequently read data, and when the block cache is full, recently accessed material is evicted.
- MemStore- This is the write cache, which keeps track of new data that hasn't yet been written to disc. A MemStore exists for each column family in a region.

- Write Ahead Log (WALs a file that keeps temporary data that isn't persisted to a permanent storage location.
- HFile is the storage file that contains the rows on a disc as sorted key values.

11.12 ZOOKEEPER

Zookeeper is an open-source project that provides services like maintaining configuration information, naming, providing distributed synchronization, etc.Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers.In addition to availability, the nodes are also used to track server failures or network partitions.Clients communicate with region servers via zookeeper.In pseudo and standalone modes, HBase itself will take care of zookeeper.

In HBase, it's similar to a coordinator. It offers features such as configuration information management, naming, distributed synchronization, and server failure notification, among others. Clients use zookeeper to communicate with region servers.Can store large data sets,Database can be shared, Cost-effective from gigabytes to petabytes, High availability through failover and replication. No support SQL structure, no transaction support, Sorted only on key, Memory issues on the cluster

Comparison between HBase and HDFS:

HBase provides low latency access while HDFS provides high latency operations.
HBase supports random read and writes while HDFS supports Write once Read Many times.
HBase is accessed through shell commands, Java API, REST, Avro or Thrift API while HDFS is accessed through MapReduce jobs.
HBase is widely used for online analytical processes, such as real-time data updates in ATM machines, and it can also be used in financial applications.

11.13 IBM Infosphere Streams

Every day, individuals and organizations generate data at a rate that would have appeared unheard of only a few years ago. For example, AT&T's network transfers roughly 30 petabytes of data each day on a typical day.

1. Every second, the Dublin City Centre analyses 50 bus location updates.

2 Every day, Twitter receives 340 million tweets, the majority of which come from mobile users.

3 Annual Internet traffic is predicted to exceed 1.3 zettabytes by 2016,

4 with unstructured data accounting for 80% of that.

Today's enterprises are pushed to make informed, real-time business decisions and stay ahead of the competition in the face of this huge volume of continuously changing data. Smart firms, on the other hand, are quickly understanding that by updating their data mining processes to handle non-traditional, unstructured data sources like audio, video, and email, they can extend the value of their existing systems and generate major commercial advantages. This allows them to respond more swiftly to shifts in client sentiment, discover new market opportunities, and launch ground-breaking new goods that are in line with current trends.

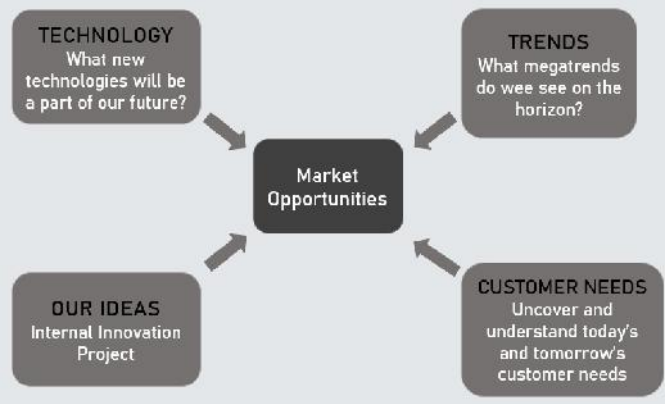


Figure 16 Market Opportunities

As a key enabler for this new generation of analytic processing methods, IBM® InfoSphere® Streams provides a state-of-the-art computing platform that can help companies transform burgeoning data into actionable information and business insights. InfoSphere Streams is a critical component of the IBM Big Data Platform and delivers a highly scalable, agile software infrastructure to perform in-motion analytics on a wide variety of relational and non-relational data types entering the system at unprecedented volumes and speeds—and from thousands of real-time sources. With InfoSphere Streams, organizations can capture and act on key business data just in time, all the time.

A new paradigm for information processing

InfoSphere Streams is the outcome of IBM Research's groundbreaking work in collaboration with the US government.Theprogramme includes a development platform and runtime environment that allows businesses to create and run applications that ingest, filter, analyse, and correlate huge amounts of data in real time.

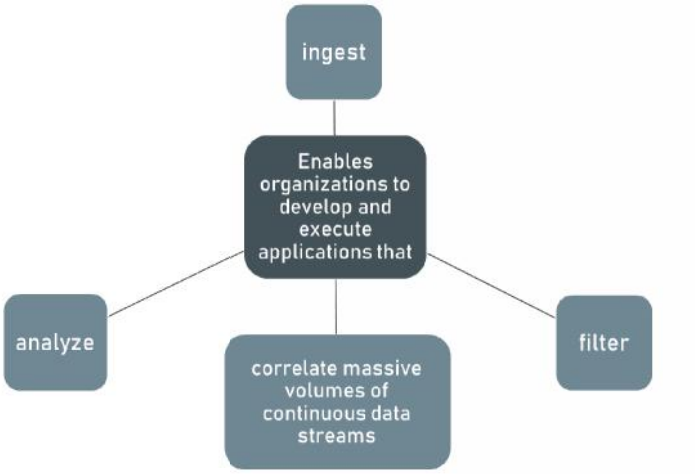


Figure 17 A new paradigm for information processing

These data streams can come from both structured and unstructured data sources, and they can contain a wide range of digital data, including:

- Text files, spreadsheets, images, video and audio recordings •
- Email, chat and instant messages, web traffic, blogs and social networking sites •
- Financial transactions, customer service records, telephone usage records, system and application logs. Data from satellites, GPS tracking, smart devices and network traffic sensors

- InfoSphere Streams brings these disparate data kinds together on a computing platform that allows for advanced data analysis while maintaining great speed and response times.

Powerful, real-time analytic processing made simple

InfoSphere Streams solves a significant data problem: analysing massive volumes of data in motion. The ability to analyse this continuously generated data is often critical for organisations that must react in real time to market alerts or events, or when large amounts of data must be filtered to identify rich, high-value information before being moved into a data warehouse or Apache Hadoop system.InfoSphere Streams use cutting-edge technologies, such as its Streams Processing Language, to analyse streaming data efficiently (SPL). To reuse existing functionality and reduce time-to-value, InfoSphere Streams applications can be enhanced with C, C++, or Java programmesFigure 18.

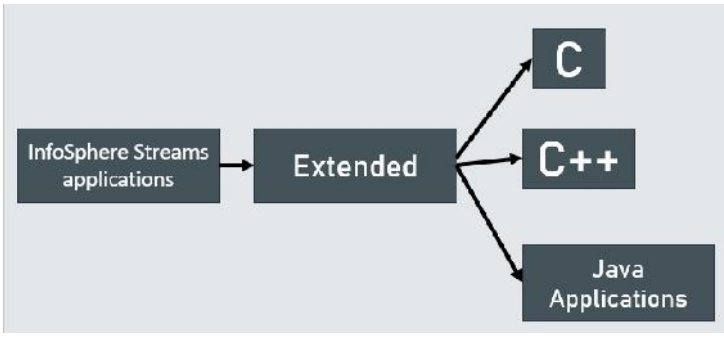


Figure 18InfoSphere Streams Applications

Getting Started also makes it simple to install, build, configure, and manage application instances with just a few clicks. Visual programming with drag-and-drop helps to shorten the learning curve and accelerate application development.

Enterprise integration: Extending the power of InfoSphere Streams

InfoSphere Streams has out-of-the-box interaction with other IBM Big Data Platform products, as well as built-in XML compatibility, allowing it to access all of your data and systems.

Figure 19 Enterprise Integration

Scale-out architecture

InfoSphere Streams software helps organizations extend their current IT investments without a massive infrastructure overhaul. It scales from a single server to a virtually unlimited number of nodes to process data of any volume—from terabytes to zettabytes. InfoSphere Streams provides a clustered runtime environment that can easily handle up to millions of events per second with microsecond latency. Actionable results can be achieved with near-zero latency. For improved speed, the Advanced Compiler combines parts of the application and can distribute parts of the application across many hardware nodes.Ethernet and InfiniBand are among the high-speed transports it supports. Existing applications can be flexibly extended with new apps that access the same data streams, allowing current investments to be used even further.A web-based management console makes it easy to configure and manage the runtime and applications, including automatically placing features and deploying application components. Applications and

their individual elements can be monitored for status and performance metrics to help ensure the company attains its service-level agreements.

11.14 Comprehensive Tools for an Agile Development Environment

Developers and administrators may quickly create applications that analyse high-volume, high-velocity, and high-variety data in real time with InfoSphere Streams.These solutions are meant to assist enterprises rapidly and easily add powerful real-time data analytics to their applications:

• *InfoSphere Streams Debugger*

InfoSphere Streams Debugger is an interactive debugger for debugging SPL applications that examines the data flow in and out of SPL operators.

The drag-and-drop graphical editor

Users can construct applications with the drag-and-drop graphical editor while the graphical and SPL source code views are immediately synced. This approach allows developers to construct an application using either a text or graphical editor, then update it later using the other editor.

An instance graph

- A visual monitoring of application health and metrics is provided by an instance graph, which is available in both InfoSphere Streams Studio and the management console, and allows users to immediately spot issues using customisable views.

The latest InfoSphere Streams data visualization capabilities

- Users can dynamically add new views to live applications using the latest InfoSphere Streams data visualisation features, which include charts that are given out-of-the-box.

Sophisticated analytics with toolkits and accelerators

InfoSphere Streams includes integrated toolkits and sample apps to make creating solutions for certain industries or services easier. The Data Mining Toolkit, Financial Services Toolkit, and Standard Toolkit, for example, contain a collection of the most often used operators for seamless interaction with existing infrastructures and applications.A financial services company, for example, may use InfoSphere Streams to integrate incoming stock market quotes with industry statistics or information on interested traders stored in an internal database. Adding context to stock market ticks allows the company to perform more comprehensive research and tailor alerts than ever before. . Furthermore, Infosphere Streams assists in determining whether data should be kept in a database for future analysis in real time, lowering storage and administration expenses.

InfoSphere Streams: System requirements

Hardware requirements

- Intel/AMD x86 architecture (64-bit) or IBM POWER7® architecture systems (minimum 500 MB memory)
  2 GB memory to run simple applications

Software requirements

- Red Hat Enterprise Linux (RHEL) Version 5.6 or later, or Version 6.1 or later for x86 architecture hardware.
- RHEL Version 6.2 and above for IBM POWER7 architecture hardware.
- CentOS Version 6.1 or later for x86 architecture hardware.
- SUSE Linux Enterprise Server (SLES) V11.2 or above for x86 architecture hardware (SLES does not support Security Enhanced Linux)
- Eclipse Platform SDK v3.6.2, v3.7.2 or v3.8.0.
- IBM Java SE v6.0-9.0 SDK for x86 or IBM POWER7 architecture hardware.
- Oracle Java SE v6.0.x SDK for x86 architecture hardware.
- Mozilla Firefox v10 and above • Microsoft Internet Explorer 9.0 and above.

Other toolkits in Infosphere Streams include:

Complex Event Processing (CEP), which uses patterns to detect composite events in streams of basic events, resulting in high performance and rich analytics. Existing applications can be simply moved to an Infosphere Streams environment to benefit from greater scalability and the capacity to handle up to 10 times more events per second on the same hardware.

Figure 20 Other Toolkits in Infosphere Streams

Summary

- Hive is a SQL-based database that allows users to read, write, and manage petabytes of data. Hive is based on Apache Hadoop, an open-source system for storing and processing massive information effectively.
- Pig is a high-level programming language for Apache Hadoop. Pig allows data analysts to create complicated data transformations without needing to know Java.
- The IBM InfoSphere Information Server is a prominent data integration technology that makes understanding, cleansing, monitoring, and transforming data easier.
- HDFS is a distributed file system that runs on commodity hardware and can handle massive data collections. It is used to grow an Apache Hadoop cluster from a few nodes to hundreds (or even thousands) of nodes. HDFS is one of Apache Hadoop's primary components, along with MapReduce and YARN.
- Hadoop includes a RecordReader that transforms input splits into key-value pairs using TextInputFormat. In the mapping process, the key-value pairs are utilised as inputs. The only data format that a mapper can read and understand is this one.
- Hadoop applications use the Hadoop Distributed File Solution (HDFS) as their primary data storage system. HDFS is a distributed file system that uses a NameNode and DataNode architecture to allow high-performance data access across highly scalable Hadoop clusters.
- Hadoop is an Apache Foundation open source framework for processing huge volumes of heterogeneous data sets in a distributed way across clusters of commodity computers and hardware using a simplified programming style. Hadoop is a dependable distributed storage and analysis solution.
- If a job fails, Hadoop will identify the failure and reschedule replacements on healthy computers. It will only end the task if it fails four times, which is the default setting that may be changed, and it will kill terminate the job. to be finished
- Namenode stores information about all other nodes in the Hadoop Cluster, files in the cluster, file component blocks and their positions in the cluster, and other information that is necessary for the Hadoop Cluster's functioning.

- Job Tracker manages the sharing of information and outcomes by keeping track of the specific tasks/jobs allocated to each of the nodes.
- The IBM InfoSphere Information Server is a prominent data integration technology that makes understanding, cleansing, monitoring, and transforming data easier.

Keywords

Apache Hive: Hive is a data warehousing and ETL solution built on top of the Hadoop Distributed File System (HDFS). Hive makes it simple to carry out tasks such as these.Encapsulation of data, Querying on the fly, Large-scale data analysis

Apache Pig: Pig is a high-level platform or tool for processing massive datasets. It provides a high-level of abstraction for MapReduce computation.

Apache MRUnit: Apache MRUnit TM is a Java package for unit testing Apache Hadoop map reduce tasks. The post's example uses the Weather dataset, and it works with the year and temperature retrieved from it. Obviously, you can simply adapt the example to your own data.

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

HBase: HBase is a data model that looks a lot like Google's Big Table. It is a Java-based open-source distributed database created by the Apache Software Foundation. HBase is a critical component of the Hadoop ecosystem. HDFS is the foundation for HBase.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Meta-Store: Apache Hive metadata is stored in Meta-store, a central repository. It uses a relational database to store Hive table and partition metadata (such as schema and location).

Name node: The name node is a piece of commodity hardware that houses the GNU/Linux operating system as well as name node software. It's a piece of software that can run on standard hardware.

Shuffling and sorting: The process of transferring intermediate output from the mapper to the reducer is known as shuffle. On the basis of reducers, a reducer receives one or more keys and their associated values. The mapper's intermediated key – value is automatically ordered by key.

Self Assessment

1. Which of the following is an open-source data warehouse system that has been built on top of Hadoop?

A. Apache Hive
B. Apache Pig
C. Apache Hbase
D. None of the mentioned

2. Which of the following are the components of Hive?

A. Driver
B. Compiler
C. Metastore
D. All of the mentioned.

3. The compiler gets the necessary metadata from the _____.
A. Meta store
B. Database
C. Management
D. All of the mentioned.

4. _____ developed by Yahoo researchers executes Map Reduce jobs on extensive datasets and provides an easy interface for developers to process the data efficiently.
A. Apache Hive
B. Apache pig
C. Both
D. None of the mentioned

5. All the Pig Scripts are handled by the _____
A. Parser
B. Optimizer
C. Both
D. None of the mentioned

6. Which of the following compiles the optimized logical plan into a series of MapReduce jobs?
A. Parser
B. Atom
C. optimizer
D. compiles

7. Which of the following are the execution modes of Apache Pig.
A. Local mode and Shuffle mode
B. Local mode and MapReduce mode
C. Sort mode and MapReduce mode
D. None of the mentioned.

8. Which of the following are the classification of Apache Pig Operators?
A. Relational and Diagnostic
B. Sub relational and Diagnostic
C. Relational and sub diagnostic
D. None of the mentioned.

9. _____ operator is used to perform an inner, equijoin join of two or more relations based on common field values.
A. COMBINE
B. COMBINATION
C. JOIN
D. JOINING

10. Apache Hive was created by _____
A. Facebook
B. Twitter
C. Both
D. None of above

11. Which of the following are the components of Apache Hive?
A. Hive Client
B. Hive Services
C. Processing and Resource Management
D. All of the above

12. Which of the following services provided by Apache Hive?
A. Hive Server, Interpreter
B. Beeline, Hive Server
C. Hive Driver, Interpreter
D. None of the above

13. Select following components of HBase architecture.
A. HMaster
B. Region Server
C. ZooKeeper
D. All of the above

14. HBase Tables are separated into Regions _____ by row key range.
A. Vertically
B. Horizontally
C. Diagonally
D. None of the mentioned

15. Which of the following is not a component of the HDFS data node?
A. Block Cache
B. MemStore
C. HFile
D. None of the mentioned

Review Questions

1. Explain architecture of Pig.
2. Explain working of HIVE.
3. Elaborate classification of Apache Pig operators
4. What do you understand by Infosphere streams?

5. Explore Enterprise integration and concepts of scale-out architecture,

Answers for Self Assessment

1.	A	2.	D	3.	A	4.	B	5.	A
6.	D	7.	B	8.	A	9.	C	10.	A
11.	D	12.	B	13.	D	14.	B	15.	D

Further Readings

Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.

Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.

McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com

Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Manning Publications.

Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.

White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/
7. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm

Unit 12: Predictive Analytics

CONTENTS

Objectives

- learn simple linear regression and multiple linear regression.
- learn visual data analysis techniques.
- learn applications of business analytics.

Introduction

Simple linear regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables.

- Onevariable,denoted x, is regarded asthe **predictor**, **explanatory**, or **independent** variable.
- The other variable, denoted y, is regarded as the **response**, **outcome**, or **dependent** variable.
- Often, the objective is to predict the value of an output variable (or response) based on the value of an input (or predictor) variable.

When to use regression

We are frequently interested in determining the link between a number of factors. Toinvestigatepot ential correlations between pairs of data, scatterplots and scatterplot matrices can be employed. Cor relation measures the linear link between two variables, but it doesn't reveal more complicated corr elations. If the connection is curvilinear, for example, the correlation may be approaching zero.

- We also look at the graph to determine the direction of the linear relationship.

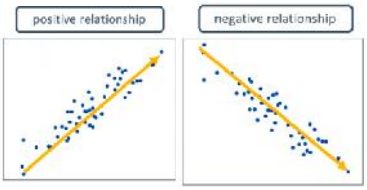


Figure 1 Positive and Negative Relationships

A line that begins in the lower left corner of the plot and ends in the upper right corner is called a positive relationship. In a positive linear relationship, high scores on the X variable predict high scores on the Y variable. A line that begins in the upper left corner of the plot and ends in the lower right corner (like the relationship shown above) is called a negative relationship as shown in Figure 1. In a negative linear relationship, high scores on the X variable predict low scores on the Y variable.

Linear Relationship

Linear regression is a **linear model**, e.g., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). Scores scattered randomly around a straight line in the middle of the graph indicate no relationship between variables.Sometimes a scatter plot will show a curvilinear relationship between two variables. If this happens, we need to use special statistics developed for curvilinear relationships.Whereas some relationships are straightforward to understand, explain, and detect statistically (i.e., linear relationships), curvilinear relationships are more complex because the nature of the relationship is different at different levels of the variables. Curvilinear relationships can occur often in communication research, given the complex, socially and contextually dependent phenomena that are the focus of such research.

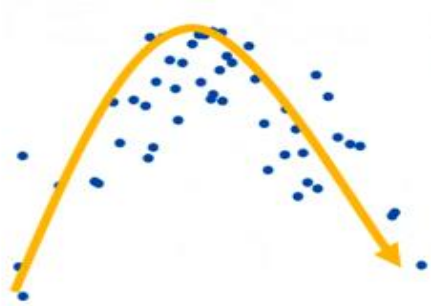


Figure 2 Curvilinear Relationship

In the context of Statistical learning, there are two types of data:

- Independent variables are data that can be directly manipulated.
- Dependent variables are data that cannot be directly controlled.

Linear regression models are a straightforward method for supervised learning. They're simple, yet they're effective.

Linear Regression

- Linear regression is nothing but a manifestation of this simple equation.

$$y = mx+c$$

- **y** is the dependent variable i.e. the variable that needs to be estimated and predicted.
- **x** is the independent variable i.e. the variable that is controllable. It is the input.
- **m** is the slope. It determines what will be the angle of the line. It is the parameter denoted as β.
- **c** is the intercept. A constant that determines the value of y when x is 0.

The same equation of a line can be re-written as:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Linear regression models are not perfect. It tries to approximate the relationship between dependent and independent variables in a straight line. Approximation leads to errors. Some errors can be reduced.Some errors are inherent in the nature of the problem. These errors cannot be eliminated. They are called as an **irreducible error**, the noise term in the true relationship that cannot fundamentally be reduced by any model.The same equation of a line can be re-written as: **β0**

and β1 are two unknown constants that represent the intercept and slope. They are the parameters. **ε** is the error term.

Example

- The following are the data provided to him: make: make of the car. fuelType: type of fuel used by the car. nDoor: number of doors. engineSize: size of the engine of the car. price: the price of the car.

Make	Fueltype	Ndoors	Enginesize	Price
alfa-romero	gas	two	130	13495
alfa-romero	gas	two	130	16500
alfa-romero	gas	two	152	16500
audi	gas	four	109	13950
audi	gas	four	136	17450
audi	gas	two	136	15250
audi	gas	four	136	17710
audi	gas	four	136	18920
audi	gas	four	131	23875

First and foremost, Fernando wants to see if he can accurately estimate automobile prices based on engine size. The following questions are addressed in the first set of analyses:

- Is price of car price related with engine size? How strong is the relationship? Is the relationship linear? Can we predict/estimate car price based on engine size?

Fernando does a correlation study.

Correlation is a metric for determining how closely two variables are linked. The correlation coefficient is a statistic that is used to quantify it. Its value ranges from 0 to 1.When the correlation coefficient is large (> 0.7), it means that if one variable grows, the other increases as well. The presence of a big -ve number implies that when one variable rises, the other falls.He does a correlation analysis. He plots the relationship between price and engine size. He splits the data into training and test set. 75% of data is used for training. Remaining is used for the test.He builds a linear regression model. He uses a statistical package to create the model. The model creates a linear equation that expresses **price of the car** as a function of **engine size.**

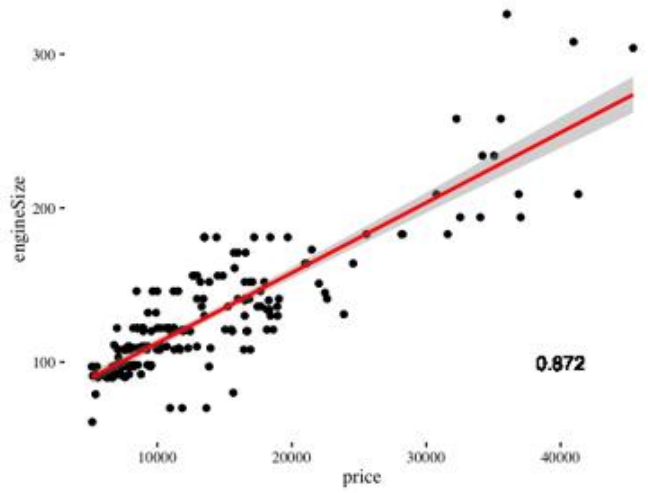


Figure 3 Model Creates a Linear Equation

Following are the answers to the questions:

- Is price of car price related with engine size? *Yes, there is a relationship.* How strong is the relationship? *The correlation coefficient is 0.872 => There is a strong relationship.*

Is the relationship linear?

- *A straight line can fit => A decent prediction of price can be made using engine size.*

Can we predict/estimate the car price based on engine size?

- *Yes, car price can be estimated based on engine size.*

12.1 Multiple Linear Regression

Multiple linear regression is used to estimate the relationship between two or more independent variables and one dependent variable. You can use multiple linear regression when you want to know:

- How strong the relationship is between two or more independent variables and one dependent variable. (e.g. how rainfall, temperature, and amount of fertilizer added affect crop growth).
- The value of the dependent variable at a certain value of the independent variables. (e.g., the expected yield of a crop at certain levels of rainfall, temperature, and fertilizer addition).

Assumptions of multiple linear regression

- **Homogeneity of variance (homoscedasticity)**: the size of the error in our prediction doesn't change significantly across the values of the independent variable. **Independence of observations**: the observations in the dataset were collected using statistically valid methods, and there are no hidden relationships among variables.
- **Independence of observations:** In multiple linear regression, it is possible that some of the independent variables are actually correlated with one another, so it is important to check these before developing the regression model. If two independent variables are too highly correlated (r2 > ~0.6), then only one of them should be used in the regression model.
- **Normality**: The data follows a normal distribution.
- **Linearity**: the line of best fit through the data points is a straight line, rather than a curve or some sort of grouping factor.

12.2 Data Visualizations

Data visualisation is one of the most essential abilities in applied statistics and machine learning.Data visualisation is one of the most significant instruments for determining a qualitative understanding. This might be useful when trying to examine a dataset and extract information about it, as well as identifying patterns, corrupt data, outliers, and other things.Data visualisations may be used to communicate and highlight significant relationships in plots and charts that are more useful to yourself and stakeholders than measurements of association or significance if we have some domain expertise.

- In today's world, we have a lot of data in our hands, thus data visualisation tools and technologies are essential for analysing vast volumes of data and making data-driven choices.It's utilised in a variety of applications, including:To create a model for complicated occurrences.Visualize phenomena that you can't see directly, such weather patterns, medical problems, or mathematical correlations.

Benefits of Good Data Visualization

- Our culture is visual, including everything from art and commercials to television and movies, since our eyes can capture colours and patterns. As a result, we can easily distinguish the red section from the blue, the square from the circular.
- As a result, data visualisation is another form of visual art that piques our curiosity and keeps our attention on the information sent via the eyes.
- We rapidly detect the patterns and outliers in the dataset whenever we show a graphic.

The following are some of the most common applications of the Data Visualization technique:

- It is a strong approach for analysing data and producing presentable and understandable findings.
- It is a fundamental stage in the pre-processing section of the data mining process.
- It aids in data cleansing by detecting inaccurate data, corrupted or missing values
- It also aids in the construction and selection of variables, which involves deciding which variables to include and exclude from the study.
- It also plays an important part in the Data Reduction process when merging the categories.

12.3 Different Types of Analysis for Data Visualization

In general, there are three types of data visualization analysis:

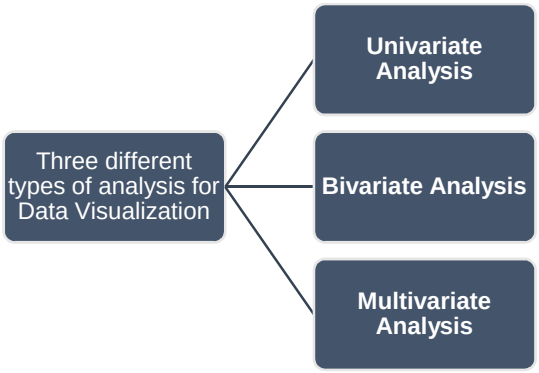


Figure 4 Different types of analysis for data visualization

Univariate Analysis: We will use a single characteristic to examine virtually all of its features in a univariate analysis. Example of univariate analysis are distribution plot, box and whisker plot and violin plot.

Figure 5 Univariate analysis techniques for data visualization

Bivariate Analysis: Bivariate analysis is when we compare data between two characteristics that are precisely the same. Example of bivariate analysis are line plot, bar plot and scatter plot.

Multivariate Analysis: We shall compare more than two variables in the multivariate analysis. For example, if a marketer wishes to compare the popularity of four adverts on a website, click rates for both men and women may be monitored, and associations between factors could be investigated.

It's similar to bivariate, however there are more dependent variables. The methods for analysing this data are determined by the objectives to be met. Regression analysis, path analysis, factor analysis, and multivariate analysis of variance are some of the approaches (MANOVA).

12.4 Applications of Predictive Analytics

Predictive analytics is the process of extracting information from data sets in order to uncover correlations, detect patterns, anticipate trends, and find linkages, among other things. This enables us to foresee the future and make the best judgments possible. Predictive analytics has several uses in corporate intelligence. We'll go through a few of the more frequent ones in this piece.

Figure 6 Applications of Predictive Analytics

- ***Customer Targeting:***Customer targeting is the process of segmenting a customer base into groups of people that are similar in characteristics that matter to marketing, such as age, gender, hobbies, and spending patterns. It allows businesses to send personalised marketing messages to people who are most likely to purchase their goods. Predictive analytics has been shown to be far more effective than traditional methods in identifying new consumers. The following are some examples of consumer targetingfactors:
  - **Socio demographic factors**:Age, occupation, marital status, education... are all socio-demographic variables.

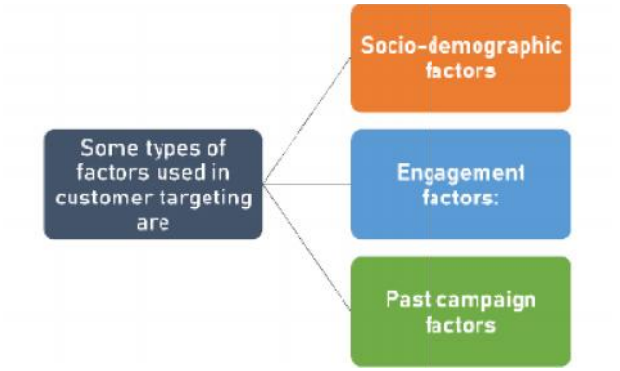


Figure 7 Factors used in customer targeting

- **Engagement factors**Factors that influence engagement include recency, frequency, and monetary value.
- **Past campaign factors**: Factors in previous campaigns: contact type, day, month, length...The benefits to the firm include I improved customer communication, (ii) significant cost savings in marketing, and (iii) a significant rise in profitability. A good example here is a banking institution's direct marketing initiatives. The objective is to forecast which customers will sign up for a term deposit.

The benefits to the firm include

(i) improved customer communication,

(ii) significant cost savings in marketing, and

(iii) a significant rise in profitability.A good example here is a banking institution's direct marketing initiatives. The objective is to forecast which customers will sign up for a term deposit.

Churn Prevention

Churn prevention tries to anticipate which consumers will leave our firm, when they will leave, and why they will go.Retaining a current client is significantly less expensive than gaining a new one. As a result, this occurrence has the potential to be extremely expensive.Companies may create predictive models that enable preemptive action before it's too late by using the power of large consumer data sets.

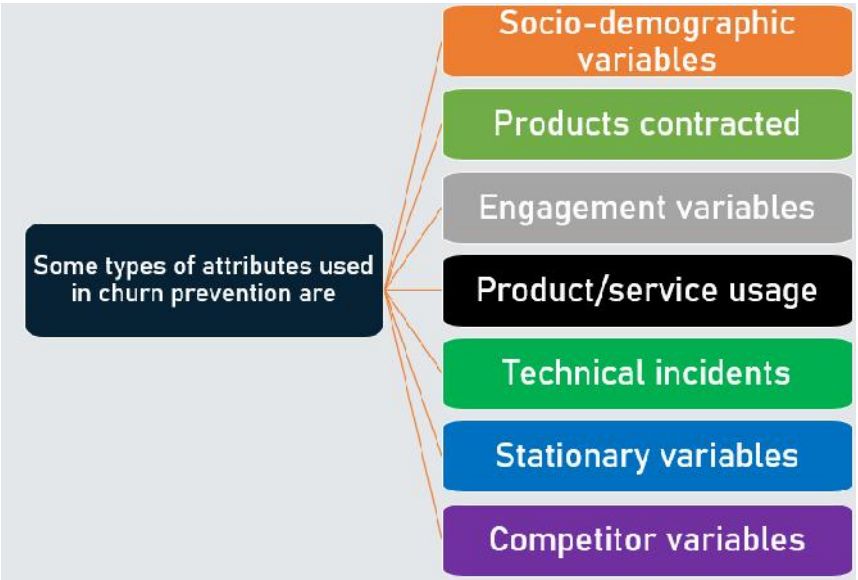


Figure 8 Attributes used in churn prevention

Sales Forecasting:

Sales forecasting examines past performance, seasonality, market-moving events, and other factors to produce a reasonable estimate of demand for a product or service. It may be used to forecast in the short, medium, or long future.Predictive analytics can forecast consumer reaction and shifting sentiments by looking at all aspects in this regard.The following are some examples of variables used in sales forecasting:Data from the calendar: season, hour, holidays, and so on.Temperature, humidity, rainfall, and other weather dataPrice, promotions, and marketing efforts are examples of company data.Economic and political aspects that a country is experimenting with are referred to as social data.

Market Analysis

Market survey analysis aids businesses in meeting consumer needs, boosting profits and lowering attrition rates.The following are some examples of quality enhancement factors:Components, presentation, etc. are some of the product's qualities.Gender, age, and other customer attributesTastes and preferences of customers are surveyed.After the firm has created the predictive model, it may use it to look for qualities that match customer preferences.For example, physicochemical testing (e.g., pH levels) can be used to predict wine quality, with the result based on sensory data (evaluations by wine experts).

Risk assessment

Risk assessment enables businesses to identify the potential for issues in their operations.Predictive analytics attempts to provide decision-making tools that can predict which activities are lucrative and which are not.Risk assessment is a broad phrase that may imply a variety of things to different people. Indeed, we may wish to assess the risk of a client, a firm, or other entity.The risk assessment can look at the following sorts of data in the instance of a client:Gender, age, education, marital status... are all socio-demographic characteristics to consider.

Financial modeling

Financial modelling is the process of converting a set of assumptions about market or agent behaviour into numerical forecasts.These prediction models are used to assist businesses in making investment and return decisions.Predicting the stock market trend using internal and external data is one example.Predictive analytics may be used to a variety of sectors and can help you improve your performance and predict future occurrences so you can respond accordingly.Neural Designer is a machine learning and data science tool that makes it simple to create prediction models.

Summary

- Regression analysis is a collection of statistical techniques used to estimate the associations between a dependent variable and one or more independent variables in statistical modelling.
- By fitting a linear equation to observed data, linear regression seeks to model the connection between two variables.
- It is a stand-alone variable that is unaffected by the other variables you're attempting to measure. A person's age, for example, might be an independent variable. Other aspects, such as what they eat, how often kids go to school, and how much television they watch, will have no effect on their age.
- In an experiment, the dependent variable is the variable that is being measured or assessed. The dependent variable in research looking at how tutoring affects test results, for example, would be the participants' test scores, because that's what's being measured.
- Correlation is a statistical word that refers to the degree to which two variables move in lockstep. When two variables move in the same direction, it is said that they have a positive correlation. A negative correlation exists when they move in opposite directions.

- The graphic depiction of data is the subject of data visualisation, which is an interdisciplinary discipline. When the data is large, such as in a time series, it is a very effective manner of communicating.
- Pig is a high-level programming language for Apache Hadoop. Pig allows data analysts to create complicated data transformations without needing to know Java.
- The phrase "bivariate analysis" refers to the study of two variables in order to discover their correlations. In quality of life research, bivariate analyses are frequently reported.
- Multivariate analysis is used to uncover patterns and relationships between several variables at the same time. Multivariate analysis is particularly effective for evaluating large datasets, since it allows you to acquire a better grasp of your data and how it connects to real-life circumstances.
- Predictive analytics is a form of advanced analytics that uses historical data, statistical modelling, data mining techniques, and machine learning to create predictions about future events. Predictive analytics is used by businesses to uncover trends in data and identify dangers and opportunities.
- The IBM InfoSphere Information Server is a prominent data integration technology that makes understanding, cleansing, monitoring, and transforming data easier.
- Hadoop includes a RecordReader that transforms input splits into key-value pairs using TextInputFormat. In the mapping process, the key-value pairs are utilised as inputs. The only data format that a mapper can read and understand is this one.

Keywords

Linear Regression: Linear regression is the process of identifying a line that best matches the data points on the plot so that we can use it to forecast output values for inputs that aren't included in the data set we have, with the assumption that those outputs will fall on the line.

Independent Variable: The independent variable (IV) is a feature of a psychological experiment that is manipulated or modified by researchers rather than by other factors..

Dependent Variable: In an experiment, the dependent variable is the variable that is being measured or assessed. The dependent variable in a research looking at how tutoring affects test results, for example, would be the participants' test scores, because that's what's being measured.

Correlation: Correlation is a statistical word that refers to the degree to which two variables move in lockstep. When two variables move in the same direction, it is said that they have a positive correlation. A negative correlation exists when they move in opposite directions.

Data Visualization:A graphical depiction of information and data is referred to as data visualisation. Data visualisation techniques make it easy to identify and comprehend trends, outliers, and patterns in data by employing visual components like charts, graphs, and maps.

Bivariate Analysis: The phrase "bivariate analysis" refers to the study of two variables in order to discover their correlations. In quality of life research, bivariate analyses are frequently reported.

Multivariate Analysis:MVA stands for multivariate analysis, which is a statistical process for analysing data including many types of measurements or observations. It might also refer to difficulties in which more than one dependent variable is investigated at the same time as other variables.

Predictive Analysis: Predictive analytics is a form of advanced analytics that uses historical data, statistical modelling, data mining techniques, and machine learning to create predictions about future events. Predictive analytics is used by businesses to uncover trends in data in order to identify dangers and opportunities.

Market Analysis: A market study is a proactive investigation of a product or service's market demand. Market research examines all of the market elements that drive demand for a particular

product or service. Price, location, competition, substitutes, and overall economic activity are all factors to consider.

HDFS: Hadoop File System was built on a distributed file system architecture. It runs on standard hardware. HDFS, unlike other distributed systems, is extremely fault-tolerant and built with low-cost hardware in mind.

Self Assessment

1. Linear regression is a _____ machine learning algorithm.
   A. Supervised
   B. Unsupervised
   C. Reinforcement
   D. Clustering

2. In Linear Regression, which of the following strategies do we apply to determine the best fit line for data?
   A. Least Square Error
   B. Maximum Likelihood
   C. Logarithmic Loss
   D. Both A and B

3. Simple linear regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables
   A. Categorical
   B. Continuous
   C. Nominal
   D. Ordinal

4. _____measures the linear link between two variables, but it doesn't reveal more complicated c orrelations.
   A. Correlation
   B. Factorization
   C. Regression
   D. None of the mentioned

5. Which of the following plot will show a curvilinear relationship between two variables?
   A. Scatter Plot
   B. Curvilinear
   C. Line
   D. Bar Plot

6. Which of the following types of variables that can be directly manipulated?
   A. Independent Variables
   B. Dependent variables
   C. Direct Variables

D. None of above

7. Which of the following types of analysis used for Data Visualization?
   A. Univariate Analysis
   B. Bivariate Analysis
   C. Multivariate Analysis
   D. All of the above

8. _____ means pictorial representation of data using graph,chart, etc.
   A. Visual data
   B. Data visualization
   C. Matplot
   D. None of the above

9. Which of the following libraries should be used to make a chart in Python?
   A. Visual data
   B. Data visualization
   C. Matplot
   D. None of the above
10. PDF stands for
   A. Probability density function
   B. Probability dendogram function
   C. Probability density fit
   D. Probability density function

11. Data visualization tools understand _____ in data
   A. Outliers
   B. Trends
   C. Patterns
   D. All of the above

12. Which of the following types of analysis used for Data Visualization?
   A. Univariate Analysis
   B. Bivariate Analysis
   C. Multivariate Analysis
   D. All of the above

13. _____ means pictorial representation of data using graph,chart, etc.
   A. Visual data
   B. Data visualization
   C. Matplot
   D. None of the above

14. Which of the following libraries should be used to make a chart in Python?
    A. Visual data
    B. Data visualization
    C. Matplot
    D. None of the above

15. PDF stands for
    A. Probability density function
    B. Probability dendogram function
    C. Probability density fit
    D. Probability density function

Answers for Self Assessment

1. What is data visualization. Explain benefits of data visualization.
2. Explain different types of analysis for data visualization.
3. Elaborate applications of predictive analytics.
4. Difference between linear regression and multiple linear regression.
5. Write note on customer targeting, churn prevention, sales forecasting, market analysis and risk assessment.

Answers for Self Assessment

1.	A	2.	A	3.	B	4.	A	5.	A
6.	A	7.	D	8.	B	9.	C	10.	D
11.	D	12.	D	13.	B	14.	C	15.	D

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
    - of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
    - White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/

2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

3. Hadoop API site: http://hadoop.apache.org/docs/current/api/

4. NoSQL databases: http://nosql-database.org/

5. Apache Spark: http://spark.apache.org/docs/latest/

6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

7. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm

Unit 13: Data Analytics with R

CONTENTS

Objectives

- learn concepts of machine learning.
- learn four categories of machine learning.

Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves.Machine Learning is the most widely used method for forecasting the future or categorizing data to assist humans in making important decisions. Machine Learning algorithms are taught over examples or situations in which they learn from previous experiences and examine historical data.When a result, as it trains over and over on the examples, it is able torecognise patterns and make predictions about the future.The learning process starts with observations or data, such as examples, direct experience, or instruction, so that we may seek for patterns in data and make better judgments in the future based on the examples we offer. The fundamental goal is for computers to learn on their own, without the need for human involvement, and to adapt their behaviour accordingly.But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, **an approach based on semantic analysis mimics the human ability to understand the meaning of a text.**

13.1 Machine Learning Methods

Machine learning approaches are traditionally divided into four broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

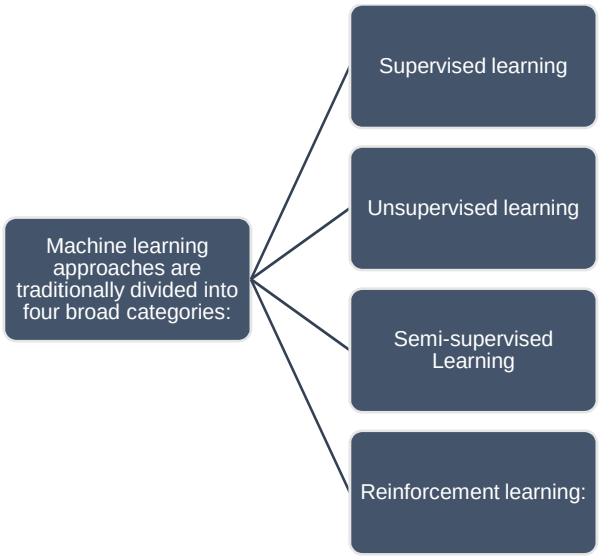


Figure 1 Machine Learning Methods

Supervised learning: Supervised machine learning algorithms can use labelled examples to apply what they've learned in the past to fresh data and predict future events. The learning algorithm creates an inferred function to generate predictions about the output values based on the examination of a known training dataset. After enough training, the system can offer objectives for any new input. The learning algorithm may also compare its output to the proper, intended output and detect mistakes, allowing the model to be modified as needed.

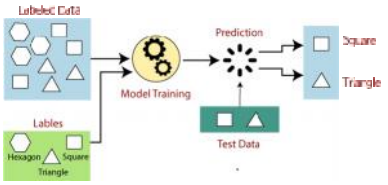


Figure 2 Supervised Learning

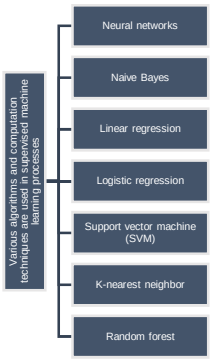


Figure 3 Supervised Learning Algorithms

- ***Neural Network:***Neural networks **reflect the behavior of the human brain**, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning.
- ***Naïve Bayes:***In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions

between the features. They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels

- ***Linear Regression***: Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.
- ***Logistic Regression:***Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- ***Support Vector Machine:***SVMs (support vector machines) are supervised machine learning techniques that may be used for both classification and regression. However, they are most commonly employed in categorization issues. SVMs were initially developed in the 1960s, but they were improved around 1990.
- ***KNN:*** The supervised machine learning method k-nearest neighbours (KNN) is a basic, easy-to-implement technique that may be used to tackle both classification and regression issues.
- ***Random Forest***: A random forest is a machine learning approach for solving classification and regression issues. It makes use of ensemble learning, which is a technique for solving difficult problems by combining many classifiers. A random forest method is made up of a large number of decision trees.

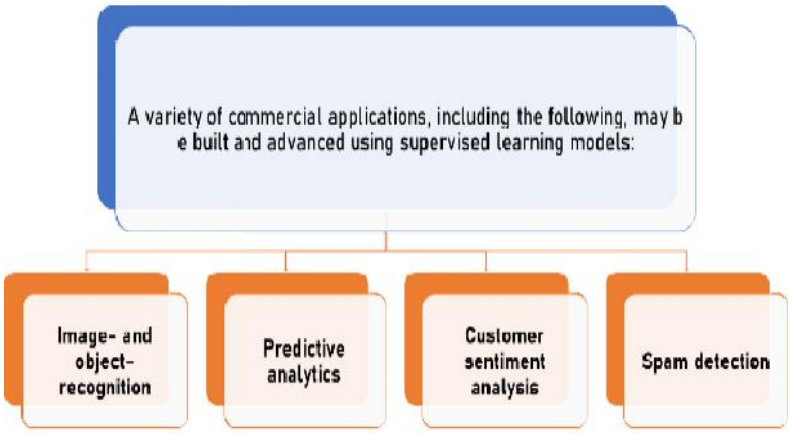


Figure 4 Supervised Learning Examples

- Image and object recognition: When applied to different computer vision techniques and visual analysis, supervised learning algorithms may be used to find, isolate, and categorise items from movies or images, making them usable.
- Predictive Analytics: The creation of predictive analytics systems to give deep insights into multiple business data points is a common use case for supervised learning models. This enables businesses to predict certain outcomes depending on a particular output variable, assisting business executives in justifying choices or pivoting for the organization's advantage.
- Customer Sentiment Analysis: Organizations can extract and categorise significant bits of information from enormous amounts of data using supervised machine learning algorithms with very little human interaction, including context, emotion, and purpose. This may be quite beneficial in terms of obtaining a better knowledge of consumer interactions and improving brand engagement initiatives.
- Spam Detection: Another example of a supervised learning model is spam detection. Organizations may train databases to identify patterns or abnormalities in fresh data using supervised classification algorithms, allowing them to efficiently arrange spam and non-spam correspondences.

13.2 Challenges of Supervised Learning

- Supervised learning models can require certain levels of expertise to structure accurately.
- Training supervised learning models can be very time intensive.
- Datasets can have a higher likelihood of human error, resulting in algorithms learning incorrectly.
- Unlike unsupervised learning models, supervised learning cannot cluster or classify data on its own.

Unsupervised Machine Learning

Unsupervised machine learning methods, on the other hand, are employed when the data being trained is neither categorised nor labelled. Unsupervised learning investigates how computers might infer a function from unlabeled data to describe a hidden structure

The system doesn't figure out the appropriate output, but it examines the data and can infer hidden structures from unlabeled data using datasets.

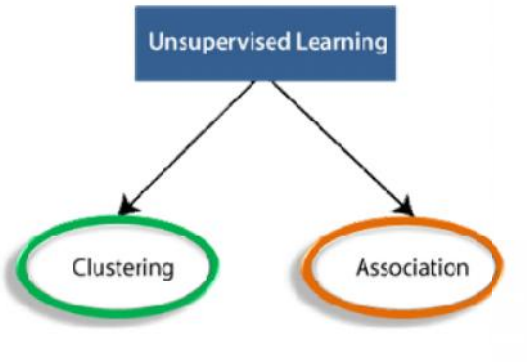


Figure 5 Unsupervised Learning

- ***Clustering***: Clustering is a way of arranging items into clusters so that those with the most similarities stay in one group while those with less or no similarities stay in another. Cluster analysis identifies similarities among data items and classifies them according to the presence or absence of such commonalities.
- ***Associate rule:*** An association rule is an unsupervised learning approach that is used to discover associations between variables in a big database. It identifies the group of items that appear in the dataset together. The association rule improves the effectiveness of marketing strategies. People who buy X (let's say a loaf of bread) are more likely to buy Y (butter/jam). Market Basket Analysis is a good example of an association rule.

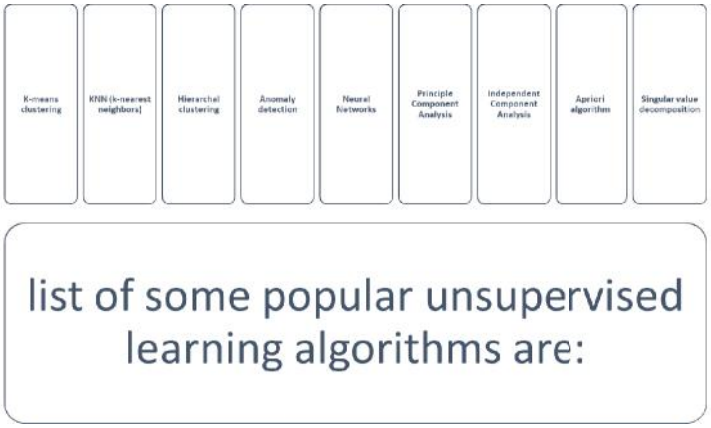


Figure 6 Popular unsupervised learning algorithms

Advantages of Unsupervised Learning

Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data. Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

Disadvantages of Unsupervised Learning

Because it lacks a comparable output, unsupervised learning is inherently more challenging than supervised learning. Because the input data is not labelled and algorithms do not know the precise output in advance, the outcome of an unsupervised learning method may be less accurate.

Semi supervised machine learning algorithms

Because they employ both labelled and unlabeled data for training – generally a small quantity of labelled data and a big amount of unlabeled data – semi-supervised machine learning algorithms lie halfway between supervised and unsupervised learning. This approach can significantly increase learning accuracy in systems that employ it. Semi-supervised learning is often used when the collected labelled data necessitates the use of competent and appropriate resources to train / learn from it. Obtaining unlabeled data, on the other hand, usually does not need additional resources.

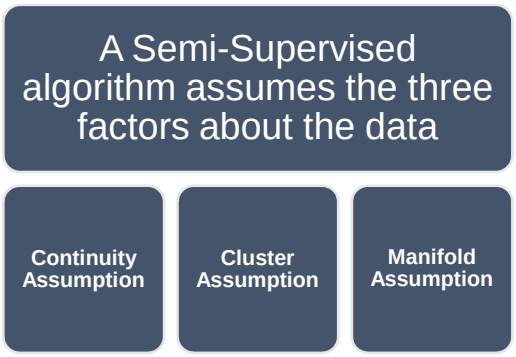


Figure 7 Three factors assume in semi-supervised learning algorithms

- A Semi-Supervised algorithm assumes three factors about the data as shown in Figure 7. **Continuity Assumption:** The method believes that points that are closer together have a higher probability of having the same output label.
- **Cluster Assumption:** The data may be split into distinct clusters, with points in the same cluster having a higher chance of having the same output label.
- **Manifold Assumption:**The data are roughly distributed over a manifold with a significantly smaller size than the input space. This assumption permits distances and densities defined on a manifold to be used.

Practical applications of Semi-Supervised Learning

Speech Analysis: Because categorising audio recordings is such a time-consuming job, semi-supervised learning is an obvious solution.

Internet Content Classification: Because labelling each webpage is difficult and impossible, semi-supervised learning techniques are used. Even Google's search algorithm ranks the relevancy of a webpage for a particular query using a form of Semi-Supervised learning.

Protein Sequence Classification: Because DNA strands are generally rather long, the emergence of semi-supervised learning in this sector has been predicted.

13.3 Reinforcement Machine Learning Algorithms

Machine Learning includes the field of reinforcement learning. It's all about taking the right steps to maximise your benefit in a given circumstance. It is used by a variety of software and computers to

determine the best feasible action or path in a given scenario. Reinforcement learning differs from supervised learning in that the solution key is included in the training data, allowing the model to be trained with the right answer, but in reinforcement learning, there is no answer and the reinforcement agent determines what to do to complete the job. It is obliged to learn from its experience in the absence of a training dataset.Here are some important terms used in Reinforcement.

Agent: It is an assumed entity which performs actions in an environment to gain some reward.

Environment (e): A scenario that an agent has to face.

Reward (R): An immediate return given to an agent when he or she performs specific action or task.

- **State (s):** State refers to the current situation returned by the environment. **Policy (π):** It is a strategy which applies by the agent to decide the next action based on the current state. **Value (V):** It is expected long-term return with discount, as compared to the short-term reward.
- **Value Function:** It specifies the value of a state that is the total amount of reward. It is an agent which should be expected beginning from that state.
- **Model of the environment:** This mimics the behavior of the environment. It helps you to make inferences to be made and also determine how the environment will behave.
- **Model based methods:** It is a method for solving reinforcement learning problems which use model-based methods. **Q value or action value (Q):** Q value is quite similar to value. The only difference between the two is that it takes an additional parameter as a current action.

How reinforcement learning works?

In this instance,Your cat is a living organism that is exposed to the outside world. It's your home in this situation. Your cat may be sitting, and you use a specific phrase in for cat to walk as an example of a condition.Our agent responds by making an action transition from one "state" to the next.

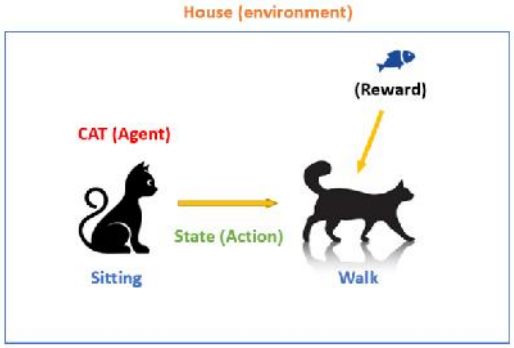


Figure 8 Example of Reinforcement Learning

Our agent responds by making an action transition from one "state" to the next.Your cat, for example, progresses from sitting to walking.An agent's reaction is an action, and a policy is a way of choosing an action given a situation in the hopes of better results.They may receive a reward or a punishment as a result of the transfer.

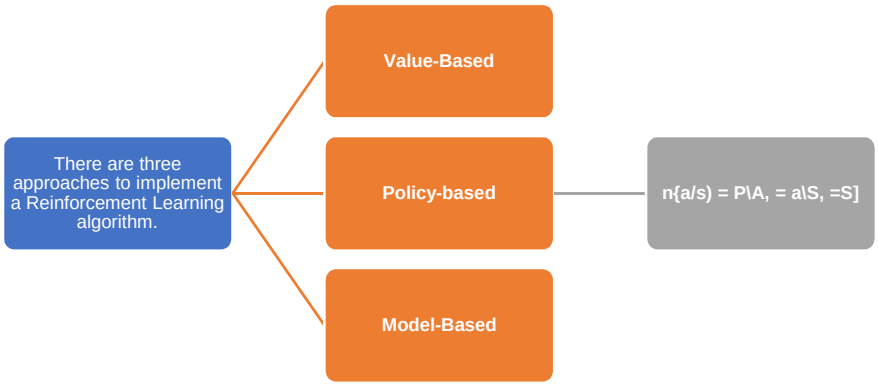


Figure 9 Three Approaches to implement reinforcement learning algorithms

- ***Value-Based***: You should aim to optimise a value function V in a value-based Reinforcement Learning technique (s). In this technique, the agent anticipates a long-term return of the existing policy states.
- ***Policy-based:***In a policy-based RL technique, you aim to come up with a policy that allows you to receive the most reward in the future by doing actions in each state.There are two sorts of policy-based methods:
  - **Deterministic:** The policy produces the same action for any state.
  - **Stochastic:** Every action has a probability, which may be calculated using the equation below.
  - **Policy of Stochasticity**: Every action has a probability.
- ***Model-based:***You must construct a virtual model for each environment in this Reinforcement Learning technique. The agent learns how to perform in that particular setting.

13.4 Characteristics of Reinforcement Learning

There is no supervisor, simply a number or a signal of reward. Making decisions in a sequential order. In Reinforcement issues, time is critical. Feedback is never immediate; it is always delayed. The data that the agent gets is determined by its activities.

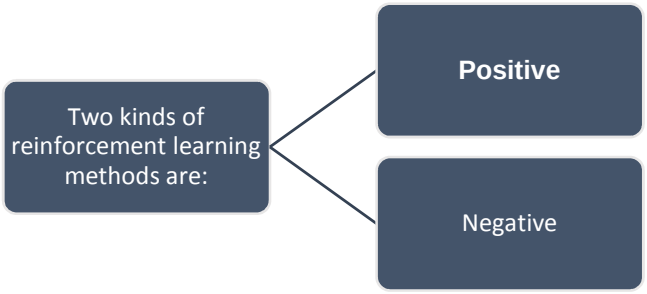


Figure 10 Kinds of Reinforcement Learning

- **Positive:**It is described as an occurrence that occurs as a result of particular actions. It improves the strength and frequency of the behaviour and has a favourable influence on the agent's actions.This sort of reinforcement aids in maximising performance and

maintaining change for a longer length of time. However, too much reinforcement might lead to state over-optimization, which can have an impact on the outcome.

- **Negative:**Bad Reinforcement is defined as behaviour strengthening that happens as a result of a negative circumstance that should have been avoided or halted. It assists you in determining the minimal level of performance. The disadvantage of this technique is that it just supplies enough to fulfil the minimal behaviour requirements.

13.5 Learning Models of Reinforcement

`There are two important learning models in reinforcement learning:

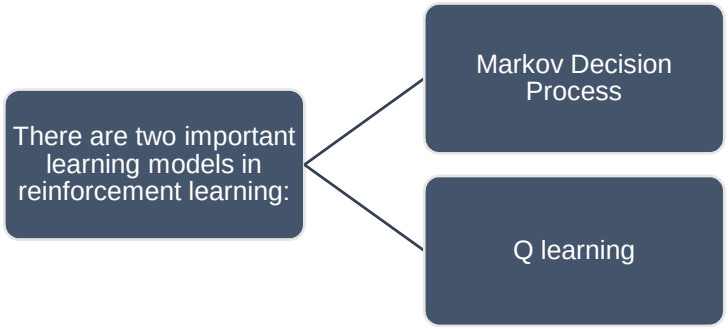


Figure 11 Important learning models in reinforcement learning

Markov Decision Process:

The following parameters are used to get a solution:

- ***Set of actions***- A, Set of states –S, Reward- R, Policy- n, Value- V. The mathematical approach for mapping a solution in reinforcement Learning is recon as a Markov Decision Process or (MDP) as shown in Figure 12.

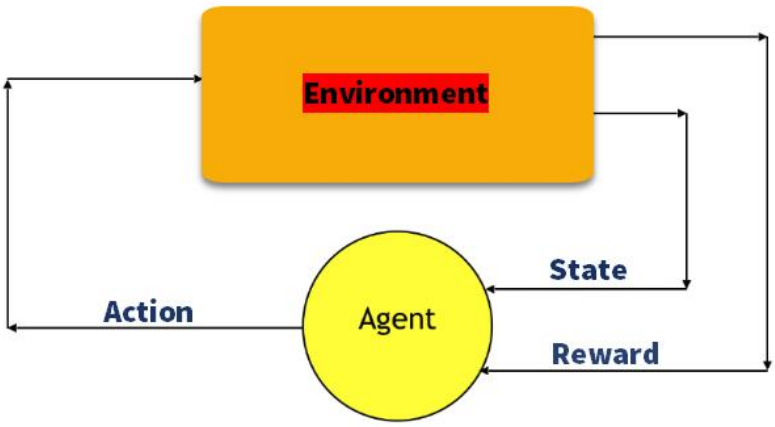


Figure 12 Set of Actions

- ***Q learning*** is a value-based approach of giving information to help an agent decide which action to perform.Let's look at an example to better understand this method:In a building, there are five rooms that are connected by doors.Each room is assigned a number from 0 to 4.The building's outside might be one large outdoor space (5)From room 5, doors 1 and 4 lead into the building.

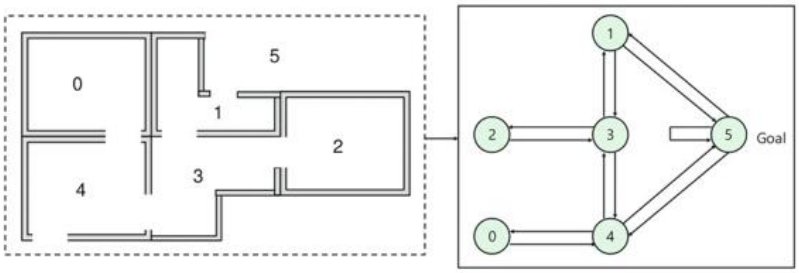


Figure 13 Q Learning

- After that, you must assign a prize value to each door:100 points are awarded for doors that go directly to the objective.There is no reward for doors that are not directly connected to the target room.Because doors are two-way and each chamber has two arrows,each of the arrows in the picture above represents an immediate prize value.

Table 1Reinforcement vs Supervised Learning

Parameters	Reinforcement Learning	Supervised Learning
Decision style	Reinforcement learning enables you to make judgments in a sequential manner.	The input supplied at the start is used to make a choice in this procedure.
Works on	Interacting with the surroundings is a priority.	Works with examples or data provided as a sample.
Dependency on decision	The learning choice in the RL technique is dependent. As a result, all of the dependent decisions should be labelled.	Supervised learning of judgments that are unrelated to one another, with labels assigned to each decision.
Best suited	Supports and works better in AI if there is a lot of human contact.	It is typically controlled by a software system or apps that are interactive.
Example	Chess game	Object recognition

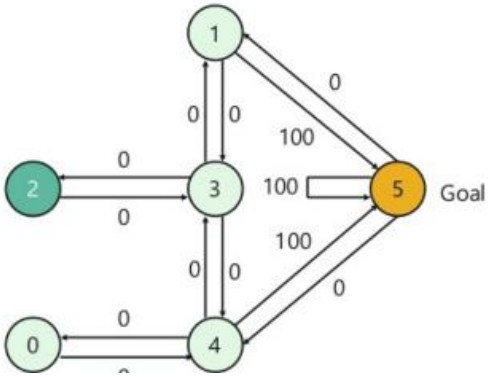


Figure 14 Q learning Assigning a prize value to each door

- In this image, you can view that room represents a state. Agent's movement from one room to another represents an action. In the below-given image, a state is described as a node, while the arrows show the action. or example, an agent traverse from room number 2 to 5
    - Initial state = state 2, State 2-> state 3, State 3 -> state (2,1,4), State 4-> state (0,5,3)
    - State 1-> state (5,3), State 0-> state 4

Applications of Reinforcement Learning

Figure 15 Robotics for industrial automation

Figure 16 Business Strategy Planning

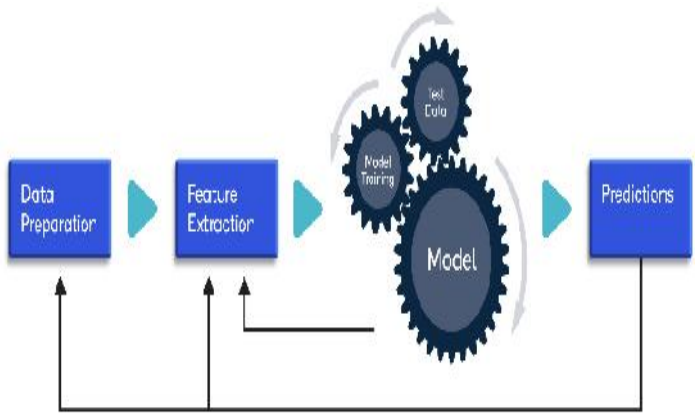


Figure 17 Machine Learning and Data Processing

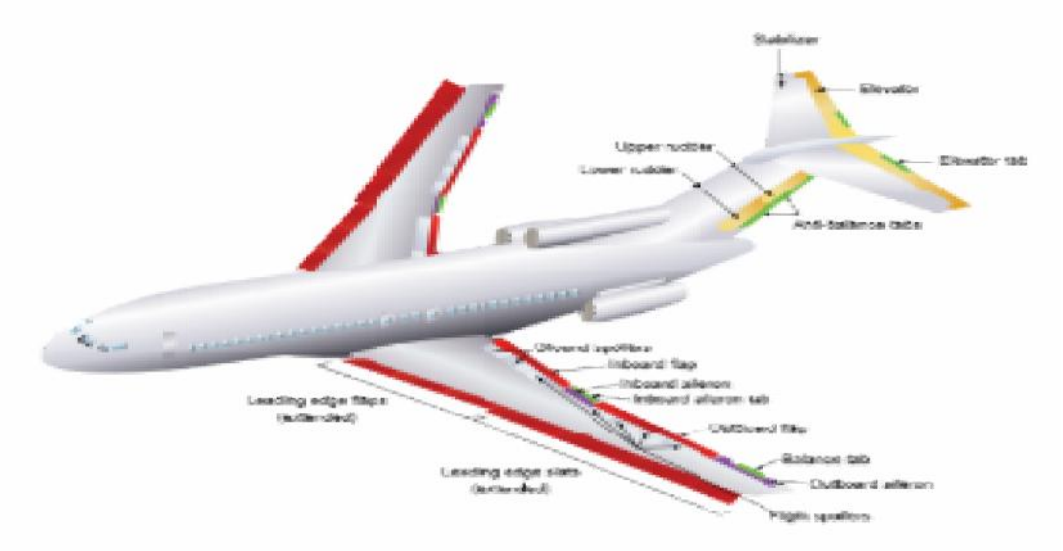


Figure 188 Applications of Reinforcement Learning

It helps you to create training systems that provide custom instruction and materials according to the requirement of students. Aircraft control and robot motion control.

Summary

- Machine learning (ML) is the study of computer algorithms that may improve themselves over time by gaining experience and using data. Machine learning algorithms create a model based on training data to make predictions or judgments without having to be explicitly programmed to do so.
- The process of supplying input data as well as proper output data to the machine learning model is known as supervised learning. A supervised learning algorithm's goal is to discover a mapping function that will translate the input variable(x) to the output variable(y) (y).
- Unsupervised learning, also known as unsupervised machine learning, analyses and clusters unlabeled information using machine learning techniques. Without the need for human interaction, these algorithms uncover hidden patterns or data groupings.

- A learning issue with a small number of labelled instances and a large number ofunlabeled examples is known as semi-supervised learning.
- Reinforcement learning (RL) is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximise the concept of cumulative reward. Reinforcement learning, along with supervised and unsupervised learning, is one of the three main machine learning paradigms.
- In statistics, naive Bayes classifiers are a subset of "probabilistic classifiers" based on Bayes' theorem and strong independence assumptions between features. They are one of the most basic Bayesian network models, but when combined with kernel density estimation, they may attain greater levels of accuracy.
- In statistics, naive Bayes classifiers are a subset of "probabilistic classifiers" based on Bayes' theorem and strong independence assumptions between features. They are one of the most basic Bayesian network models, but when combined with kernel density estimation, they may attain greater levels of accuracy.
- Sentiment analysis is the systematic identification, extraction, quantification, and study of emotional states and subjective information using natural language processing, text analysis, computational linguistics, and biometrics.
- Clustering is the process of splitting a population or set of data points into many groups so that data points in the same group are more similar than data points in other groups. To put it another way, the goal is to separate groups with similar characteristics and assign them to clusters.
- In psychology, association refers to a mental link formed by specific experiences between concepts, events, or mental states. Behaviorism, associationism, psychoanalysis, social psychology, and structuralism are all schools of thought in psychology that use associations.

Keywords

Machine Learning: Machine learning is a type of data analysis that automates the creation of analytical models. It's a field of artificial intelligence based on the premise that computers can learn from data, recognise patterns, and make judgments with little or no human input.

Linear Regression: Linear regression is the process of identifying a line that best matches the data points on the plot so that we can use it to forecast output values for inputs that aren't included in the data set we have, with the assumption that those outputs will fall on the line.

Supervised Learning: The machine learning job of learning a function that translates an input to an output based on example input-output pairs is known as supervised learning. It uses labelled training data and a collection of training examples to infer a function.

Unsupervised Learning: Unsupervised learning is a sort of algorithm that uses untagged data to discover patterns. The objective is that the machine will be pushed to create a compact internal picture of its surroundings through imitation, which is the fundamental method young infants learn, and will be able to generate inventive material as a result.

Semi-supervised Learning: Semi-supervised learning is a machine learning technique that involves training using a small quantity of labelled data and a big amount of unlabeled data. Semi-supervised learning is the middle ground between unsupervised and supervised learning. It's a unique case of poor supervision.

Reinforcement Learning: Reinforcement learning is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximise the concept of cumulative reward.

Naïve Bayes: The Bayes Theorem provides the basis for the Nave Bayes algorithm, which is utilised in a broad range of classification problems. In this essay, we will learn about the Nave Bayes algorithm as well as all of the key ideas so that there are no questions.

Clustering: Clustering is the process of splitting a population or set of data points into many groups so that data points in the same group are more similar than data points in other groups. To put it another way, the goal is to separate groups with similar characteristics and assign them to clusters.

Association analysis: The challenge of uncovering intriguing correlations in vast datasets is known as association analysis. There are two types of interesting relationships: frequent item sets and association rules. According to association rules, two objects have a strong link.

Markov Decision Process: A Markov decision process (MDP) is a discrete-time stochastic control process in mathematics. It gives a mathematical framework for modelling decision-making in settings where outcomes are partially random and partly controlled by a decision maker.

Q Learning:Q-learning is a model-free reinforcement learning technique for determining the worth of a certain action in a given state. It doesn't require an environment model, and it can handle problems like stochastic transitions and incentives without the need for adaptations.

Predictive Analysis: Predictive analytics is a form of advanced analytics that uses historical data, statistical modelling, data mining techniques, and machine learning to create predictions about future events. Predictive analytics is used by businesses to uncover trends in data in order to identify dangers and opportunities.

Market Analysis: A market study is a proactive investigation of a product or service's market demand. Market research examines all of the market elements that drive demand for a particular product or service. Price, location, competition, substitutes, and overall economic activity are all factors to consider.

Self Assessment

1. Supervised machine learning algorithms can use _____- examples to apply what they've learned in the past to fresh data and predict future events.

A. Labelled
B. Unlabelled
C. Predicted
D. Unpredictable

2. Which of the following are examples of machine learning?

A. Neural networks
B. Naïve Bayes
C. Linear Regression
D. All of the above

3. In which of the following option _____ networks reflect the behavior of the human brain, allowing computers to recognize patterns and solve common problems.

A. Neural networks
B. Naïve Bayes
C. Linear Regression
D. All of the above

4. Naive Bayes classifiers are a family of simple _____.

A. unprobabilistic classifiers

B. probabilistic central
C. probabilistic classifiers
D. None of above

5. Machine learning is an application of _____
A. Artificial Intelligence
B. Blockchain
C. Both a and b
D. None of the above

6. Which of the following techniques used in recommendation system?
A. Content based filtering
B. Collaborative filtering
C. Both
D. None of above

7. If we are considering feature to understand the taste of user that is example of _____
A. Content based filtering
B. Collaborative filtering
C. Both
D. None of above

8. In which of the following options automatic prediction is done for user.
A. Content based filtering
B. Collaborative filtering
C. Both
D. None of above

9. Collaborative filtering is an _____
   A. Supervised learning
   B. Unsupervised learning
   C. Both
   D. None of the above

10. _____ uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback.
A. Content-based filtering
B. Collaborative filtering
C. Both
D. None of the above

11. The functionality of R is divided into _____

A. Functions
B. Packages
C. Domains
D. Classes

12. Advanced users can edit R objects directly using _____ Computer code.
A. C, C++
B. C++, Java
C. Java, C
D. Java

13. In the R programming language, which of the following is utilised for statistical analysis?
A. Studio
B. Heck
C. KStudio
D. RStudio

14. Which of the following is not a data type?
A. Data frame
B. Character
C. Numeric
D. Integer

15. The R programming language resembles the __ programming language on the surface.
A. C
B. Java
C. C++
D. None of the above

Answers for Self Assessment

1.	A	2.	D	3.	A	4.	C	5.	A
6.	C	7.	A	8.	B	9.	A	10.	A
11.	B	12.	A	13.	D	14.	A	15.	A

Review Questions

1) What is machine learning? Why is the machine learning trend emerging so fast.?
2) Explain different types of machine learning algorithms.
3) Elaborate difference between classification and regression.

4) What is reinforcement learning? Explain learning models of reinforcement learning.
5) Write down difference between reinforcement learning and supervised learning.

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices
- of Scalable Realtime Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NoSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/
7. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm

Unit 14: Big Data Management using Splunk

CONTENTS

Objectives

- explore concepts of SPLUNK
- Learn features of Splunk
- Understand Interfaces, data ingestion and uploading data
- Understand concepts of data Meer
- learn steps to install Splunk enterprise on windows

Introduction

Splunk is a software used to search and analyze machine data. This machine data can come from web applications, sensors, devices or any data created by user. It serves the needs of IT infrastructure by analyzing the logs generated in various processes but it can also analyze any structured or semi-structured data with proper data modelling. It has built-in features to recognize the data types, field separators and optimize the search processes. It also provides data visualization on the search results.

Prerequisites

- The reader should be familiar with querying language like SQL.
- General knowledge in typical operations in using computer applications like storing and retrieving data and reading the logs generated by computer programs will be a highly useful.

- Splunk is a software which processes and brings out insight from machine data and other forms of big data. This machine data is generated by CPU running a webserver, IOT devices, logs from mobile apps, etc
- It is not necessary to provide this data to the end users and does not have any business meaning. However, they are extremely important to understand, monitor and optimize the performance of the machines.
- Splunk can read this unstructured, semi-structured or rarely structured data. After reading the data, it allows to search, tag, create reports and dashboards on these data. With the advent of big data, Splunk is now able to ingest big data from various sources, which may or may not be machine data and run analytics on big data.
- So, from a simple tool for log analysis, Splunk has come a long way to become a general analytical tool for unstructured machine data and various forms of big data.

14.1 Product Categories

Splunk is available in three different product categories as follows −

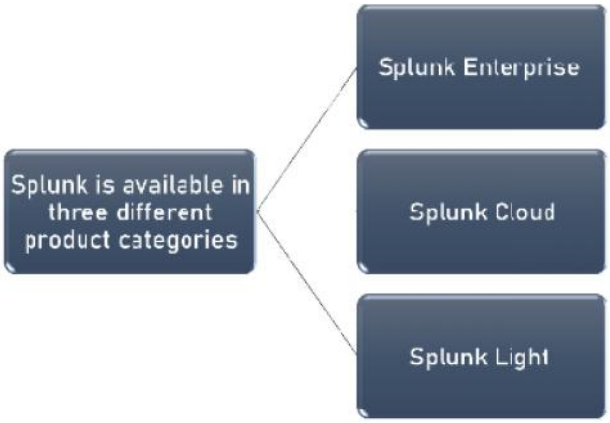


Figure 1 Three different product categories

Splunk Enterprise It is utilized by organizations with a significant IT infrastructure and a company that is heavily reliant on technology. It aids in the collection and analysis of data from websites, applications, devices, and sensors, among other sources.

Splunk Cloud It's a cloud-hosted platform with the same functionality as the corporate edition. It's available directly from Splunk or via the AWS cloud platform.

Splunk Light − It allows users to search, report, and get alerts on all log data in real time from a single location. In comparison to the other two versions, it offers less capabilities and features.

Features of SPLUNK

Features of SPLUNK are shown in Figure 2.

Figure 2 Features of SPLUNK

- ***Data Ingestion***: Splunk accepts a wide range of data types, including JSON, XML, and unstructured machine data such as web and application logs. The user can model the unstructured data into a data structure as desired.
- ***Data Indexing***: Splunk indexes the imported data for quicker searching and querying under various situations.
- ***Data Searching:***In Splunk, searching entails utilising the indexed data to create metrics, forecast future trends, and spot patterns.
- ***Using Alerts:***When certain criteria are identified in the data being examined, Splunk alerts may be used to send emails or RSS feeds.
- ***Dashboards:***Splunk Dashboards may display search results as charts, reports, and pivot tables, among other things.
- ***Data Model:***Based on specialized domain knowledge, the indexed data can be modelled into one or more data sets. This makes it easy for end users to navigate and evaluate business cases without having to grasp the intricacies of Splunk's search processing language.

14.2 SPLUNK Interface

Splunk's web interface includes all of the tools you'll need to search, report, and analyse the data you've ingested. The same web interface allows administrators to manage users and their responsibilities. It also includes connections for data intake as well as Splunk's built-in applications. The screenshot below displays the first screen you see when logging into Splunk with your admin credentials.

Administrator Link

The Administrator drop down menu allows you to customise and modify the administrator's information. Using the interface below, we may reset the admin email ID and password.

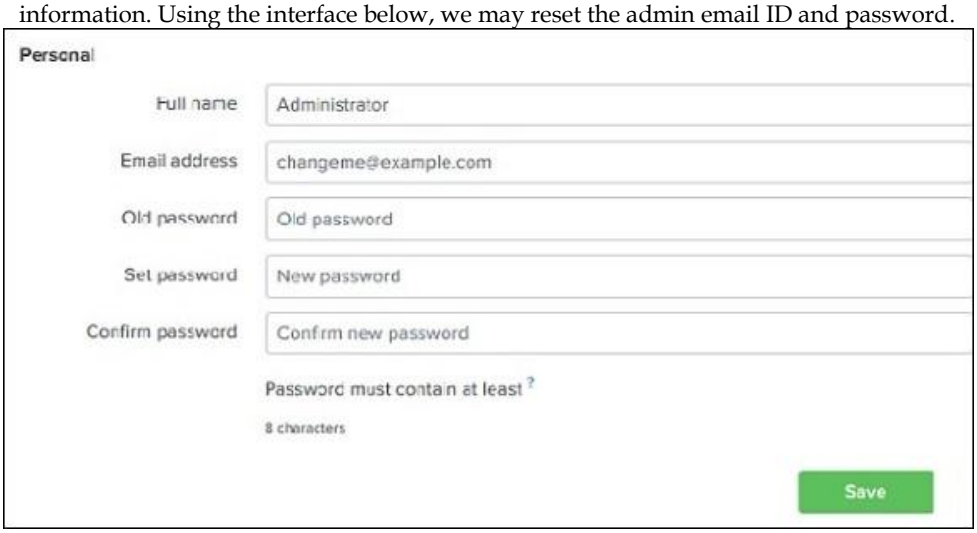


Figure 3 Administrator Link

We can also go to the preferences option from the administrator link to select the time zone and home application on which the landing page will open once you log in. It now appears on the home page, as shown below in Figure 4.

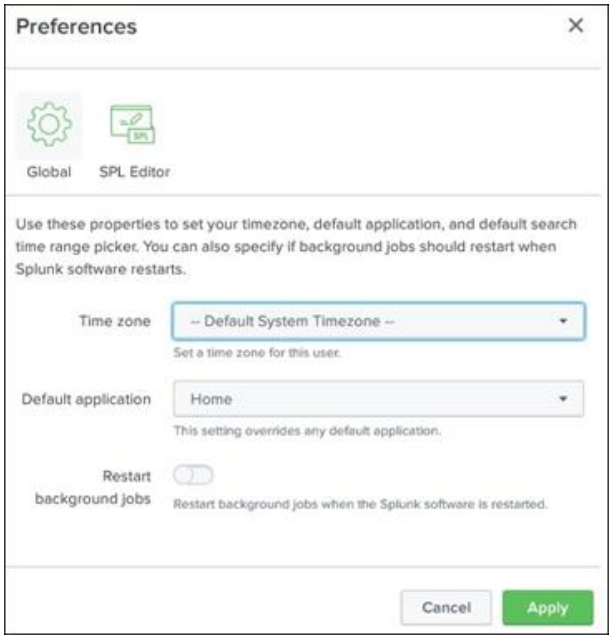


Figure 4 Preferences

Settings Link

This is a link to a page that lists all of Splunk's key functionality. By selecting the lookup link, you may add the lookup files and lookup definitions, for example.

- ***Search and Reporting Link***: The link to search and reporting brings us to the features where we can locate the data sets that are accessible for searching the reports and alerts that have been produced for these searches. The screenshot below clearly demonstrates this. −

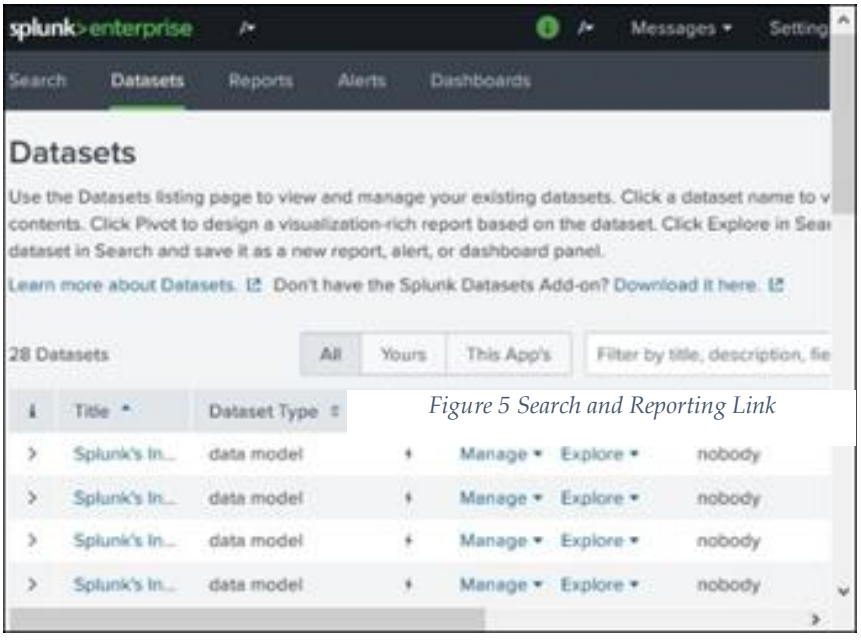


Figure 5 Search and Reporting Link

Splunk's Add Data function, which is part of the search and reporting interface, is where data gets ingested.

14.3 Data Ingestion

Data ingestion in Splunk happens through the Add Data feature which is part of the search and reporting app. After logging in, the Splunk interface home screen shows the Add Data icon.

About uploading data

When you add data to your Splunk deployment, the data is processed and transformed into a series of individual events that you can view, search, and analyze

What kind of data?

Where is the data stored?

- The process of transforming the data is called **indexing**.
- During indexing, the incoming data is processed to enable fast searching and analysis. The processed results are stored in the index as **events**.
- The index is a **flat file repository** for the data. The index resides on the computer where you access your Splunk deployment.

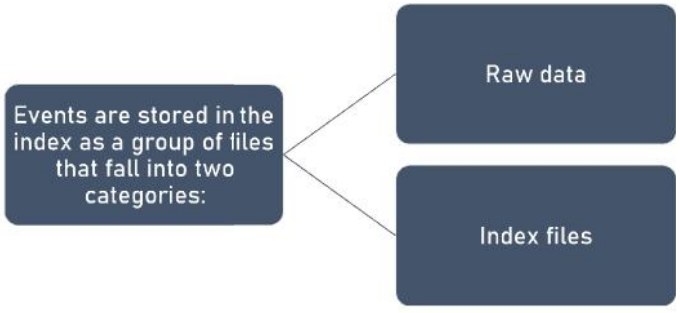


Figure 6Events are stored in the index as a group of files that fall into two categories

Use the ADD Data Wizard

Splunk's Add Data function, which is part of the search and reporting interface, is where data gets ingested. The Add Data icon appears on the Splunk interface home screen after logging in, as illustrated below.

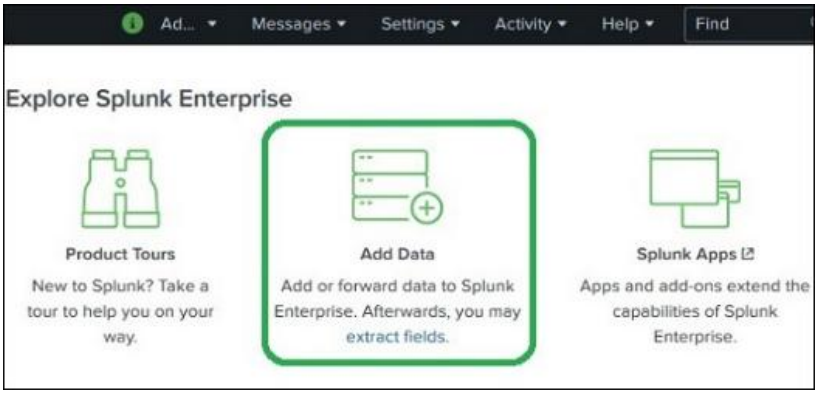


Figure 7 Add Data Wizard

When we click this option, we're sent to a screen where we can choose the source and format of the data we want to send to Splunk for analysis.

Gathering the Data

The data for analysis may be obtained from Splunk's official website. Save this file to your local disc and unzip it. When you access the folder, you'll see three files in various formats. They are log files created by some online applications. We can also get a different collection of data from Splunk, which can be found on the official Splunk website.

14.4 Uploading Data

Then, as described in the previous paragraph, we choose the file secure.log from the mailsv folder on our local machine. Using the green-colored next button in the upper right corner, we go to the next stage after selecting the file.

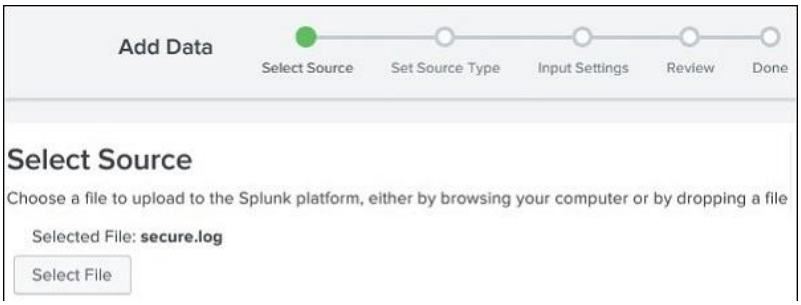


Figure 8 Uploading Data

Selecting Source Type

Splunk includes a built-in function that detects the data type being ingested. It also allows the user to select a data type other than the one suggested by Splunk. When we select a source type from the drop-down menu, we can see a list of data kinds that Splunk can ingest and search.

In the following example, we'll use the default source typeFigure 9.

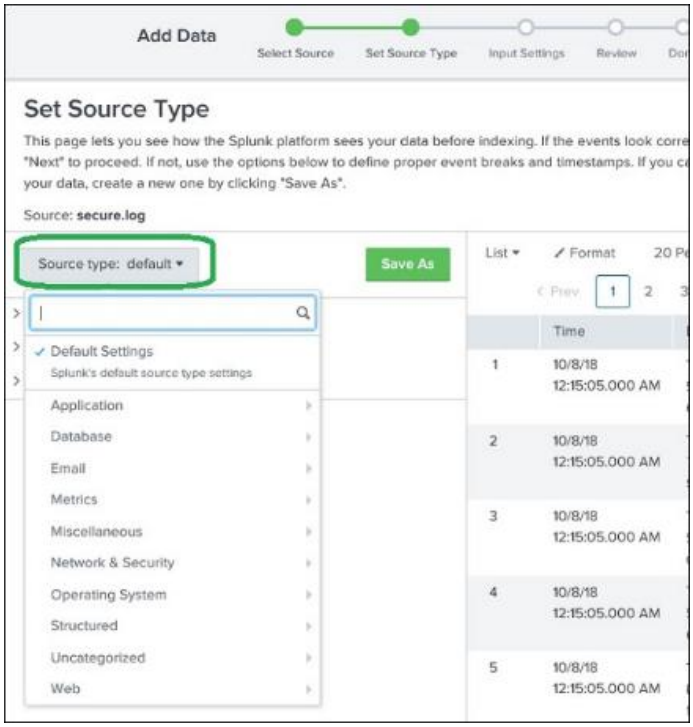


Figure 9 Set Source Type

Input Settings

We configure the host name from which the data is imported in this phase of the data ingestion process. For the host name, there are several possibilities to pick from as shown in Figure 10.

Constant value

It's the full host name of the server where the source data is stored.

regex on path

When using a regular expression to obtain the host name. Then, in the Regular expression area, type the regex for the host you wish to extract.

segment in path

Enter the segment number in the Segment number box to extract the host name from a segment in your data source's route. For example, if the source path is /var/log/ and you want the host value to be the third segment (the host server name), enter "3."

The next step is to select the index type that will be used to search the input data. The default index approach is chosen. The summary index is used to construct a summary of the data and establish an index on it, whereas the history index is used to store the search history. In the image below, it is clearly represented.

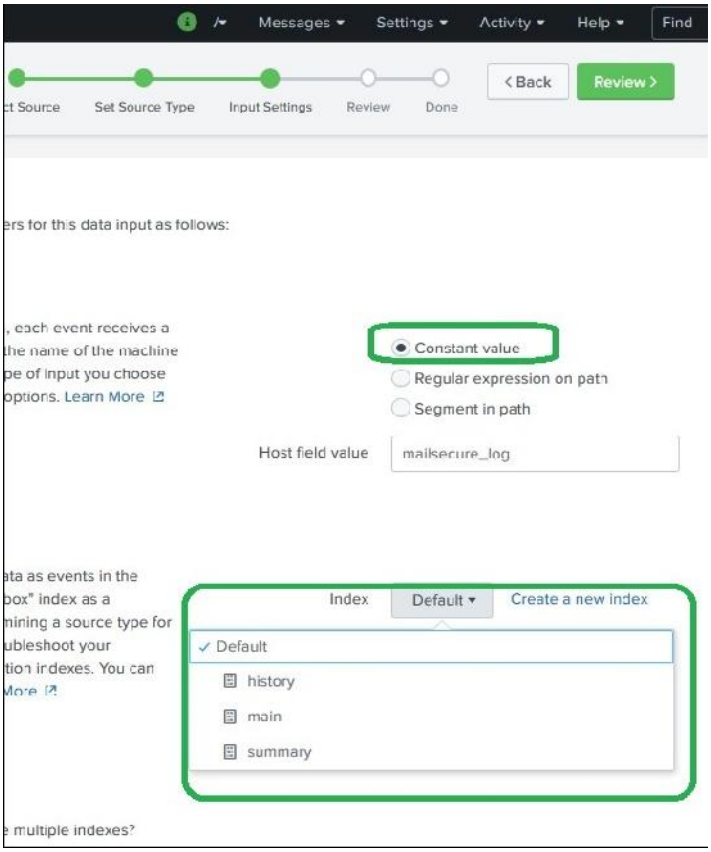


Figure 10 Input Settings

Review Settings

After clicking on the next button, we see a summary of the settings we have chosen. We review it and choose Next to finish the uploading of dataas shown in Figure 11.

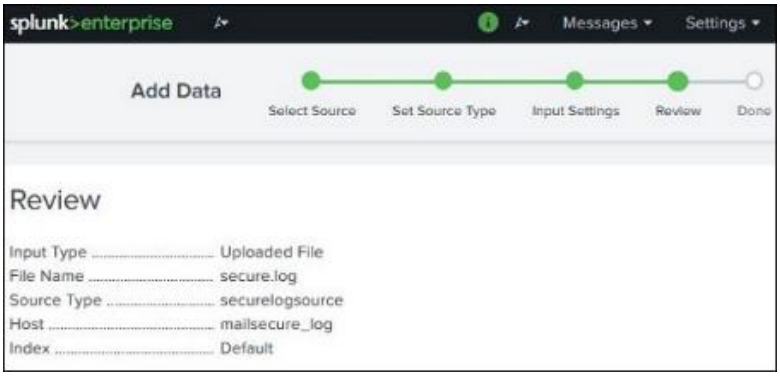


Figure 11Review Settings

When the load is complete, the screen below opens, indicating that the data was successfully ingested and outlining the next steps we may do with the data.

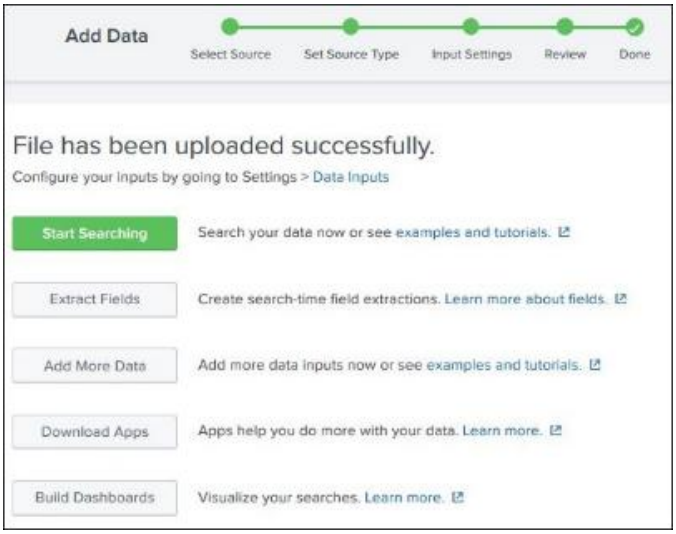


Figure 12 Data was Successfully Ingested

Splunk's inbuilt data processing unit evaluates all incoming data and classifies it into several data kinds and categories. Splunk, for example, can distinguish a log from an Apache web server and construct suitable fields from the data read.

Splunk's source type identification capability does this by utilising its built-in source types, sometimes known as "pretrained" source types.

The user does not have to manually classify the data or assign any data types to the fields of the incoming data, making analysis easy.

Supported Source Types

Uploading a file using the Add Data function and then choosing Source Type from the menu will reveal the supported source types in Splunk. We've uploaded a CSV file and then checked all of the possible choices in the Figure 13 below.

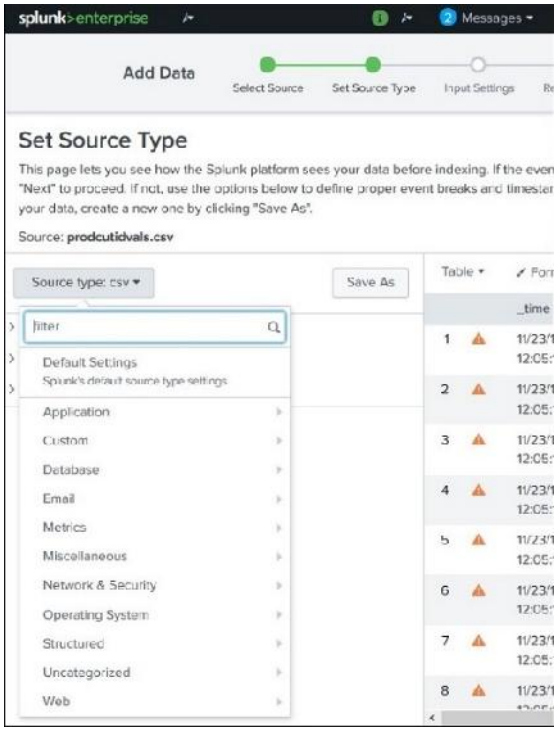


Figure 13 Add Data Feature

Source Type Sub-Category

Even inside those categories, we may click to see all of the supported subcategories. When you choose the database category, you'll be able to see the many types of databases as well as the supported files that Splunk can detect.

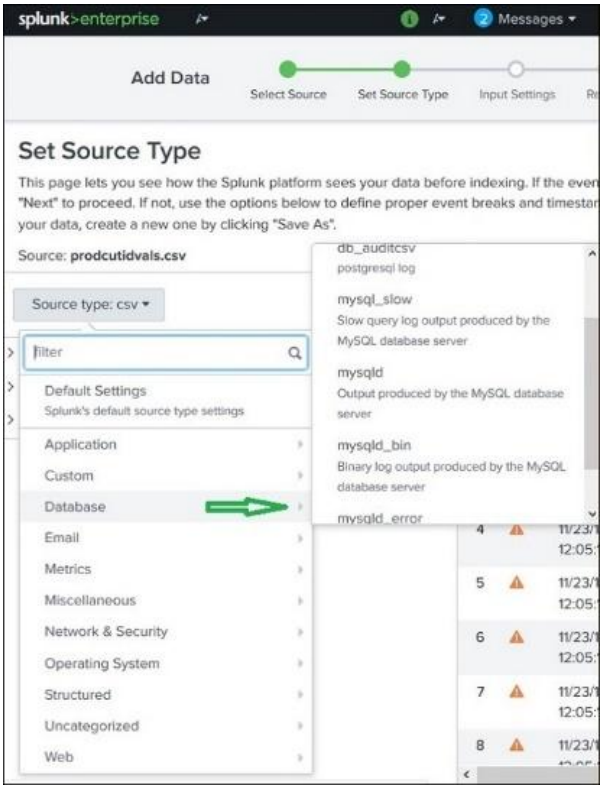


Figure 14 Source Type Sub-Category

Pre-Trained Source Types

Some of the most important pre-trained source kinds are included in the table below. Splunk is aware of this.

Table 1 Pre-Trained Source Types

Source Type Name	Nature
access_combined	NCSA's hybrid format logs from the http web server
access_combined_wcookie	http web server logs in NCSA combined format (may be created by apache or other web servers), with a cookie field added at the end
apache_error	Error log for the Apache web server
linux_messages_syslog	Standard linux syslog (on most systems, /var/log/messages)
log4j	Any J2EE server that uses log4j produces log4j standard output.
mysqld_error	Mysql's standard error log

14.5 Search & Reporting App

Splunk provides a powerful search feature that allows you to search the whole data collection that has been ingested. This functionality may be used via the Search & Reporting app, which can be found in the left sidebar after logging in to the online site.

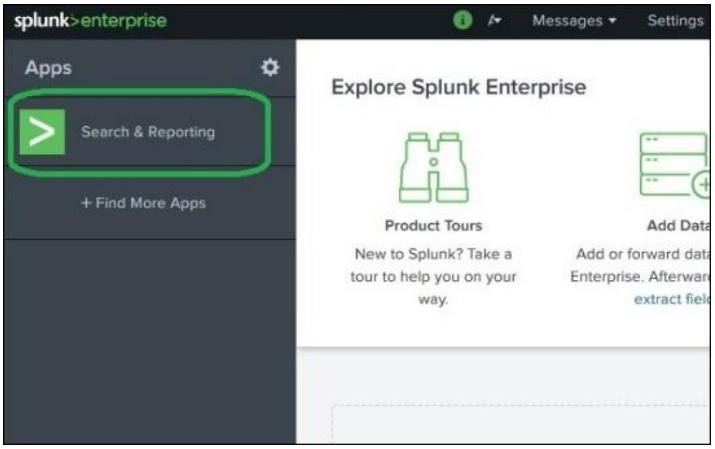


Figure 15 Search and Reporting

When we select the Search & Reporting app, we are greeted with a search box from which we can begin our log data searchFigure 16. We input the host name in the format indicated below and then click the search icon in the upper right corner. This returns a result that highlights the search word.

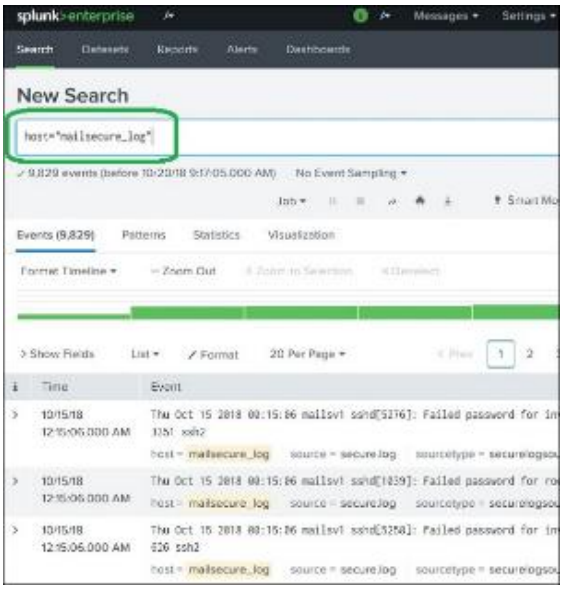


Figure 16 Search box in Search and Reporting app

Combining Search Terms

By writing the phrases one after the other, while enclosing the user search strings in double quotes, we may combine the terms used for searching as shown in Figure 17.

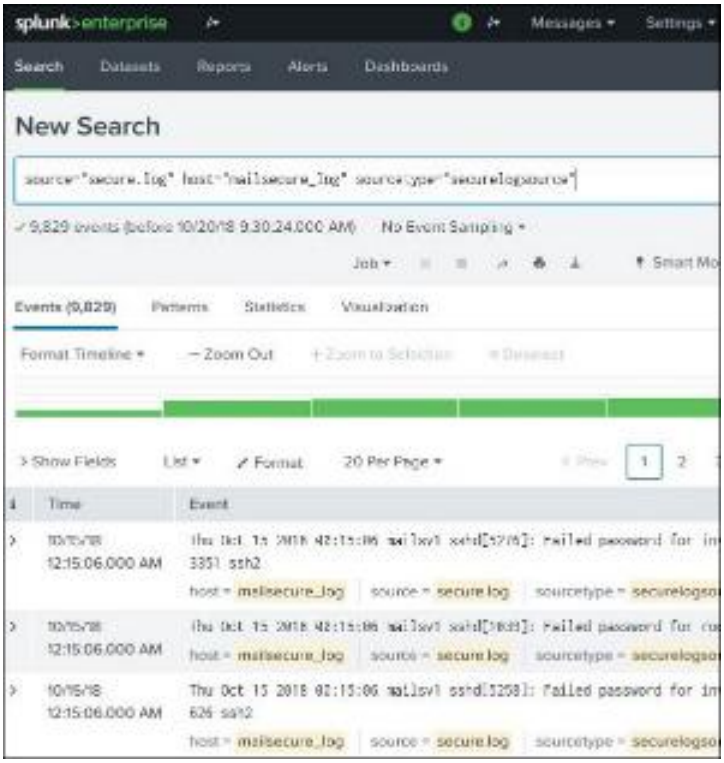


Figure 17 Combining Search Terms

Using Wild Card

We may mix wild cards with AND/OR operators in our search option. The following search yields a result in which the log file contains the phrases fail, failed, failure, etc., as well as the term password on the same line.

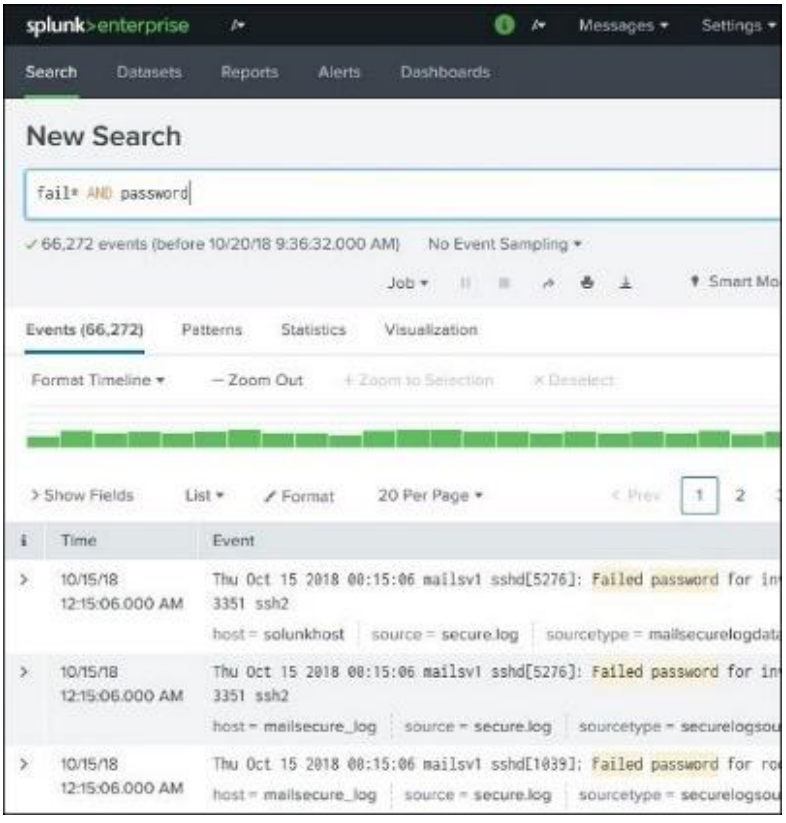


Figure 18 Using Wild Card

Refining Search Results

By picking a string and adding it to the search, we may further filter the results. In the example below, we pick the option Add to Search after hovering over the string 3351.

When we add 3351 to the search phrase, we get the following result, which only shows lines from the log that include 3351. Note how the time line of the search result has altered as the search has been narrowed as shown in Figure 19.

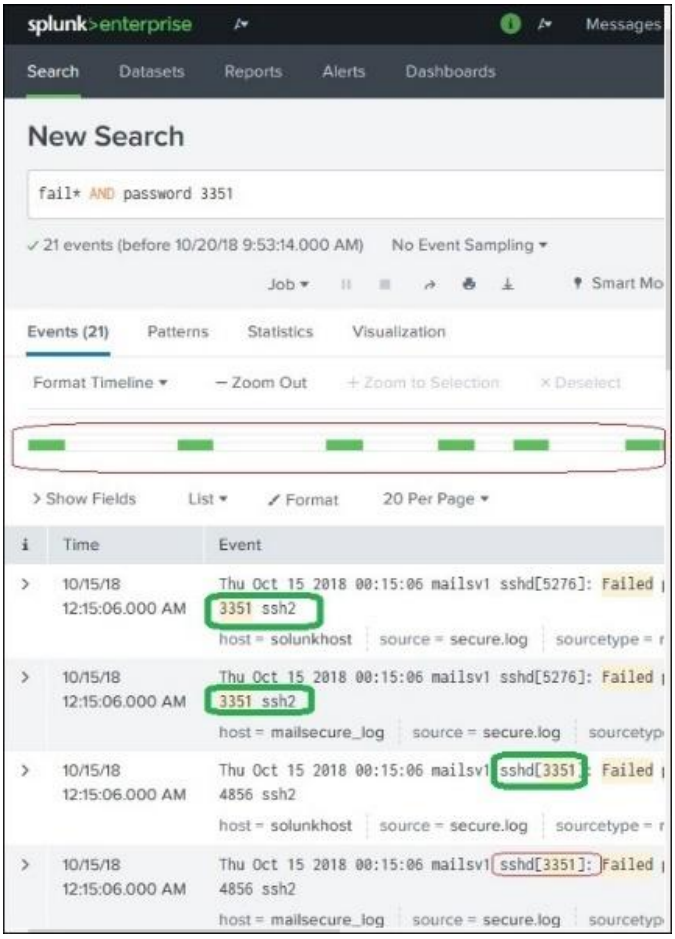


Figure 19 Refining Search Results

14.6 Splunk-Field Searching

Splunk evaluates the submitted machine data and separates it into numerous fields, each of which represents a single logical truth about the full data record.

For example, a single record of information may include the server's name, the event's date, the type of event being logged (such as a login attempt or a http response), and so on. Splunk tries to partition fields into key value pairs or separate them depending on the data types they include, such as numeric and text, even if the data is unstructured.

We can view the fields from the secure.log file by selecting the show fields option, which will bring up the page below. The fields created by Splunk from this log file are shown in Figure 20.

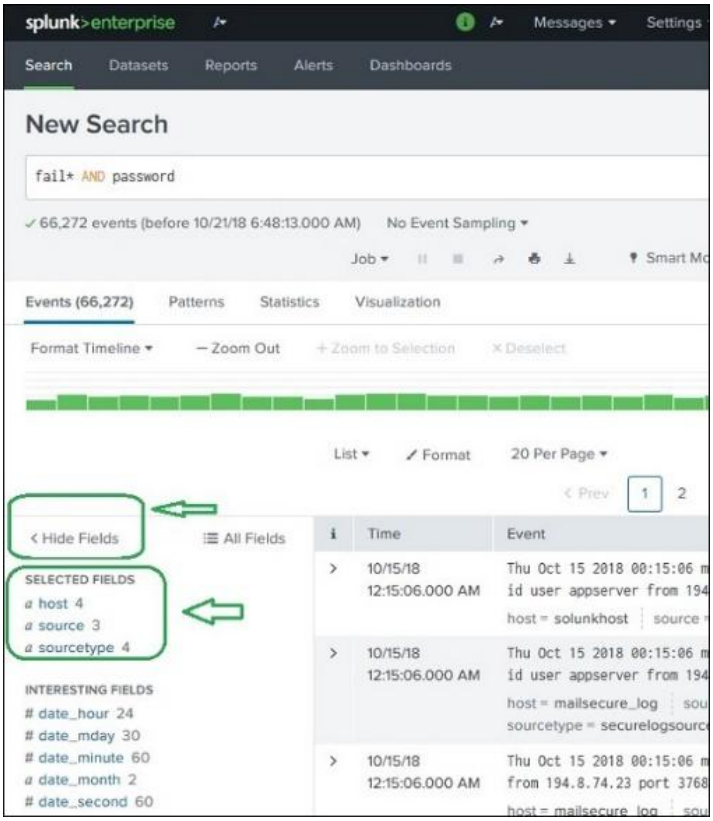


Figure 20 Field Searching

Choosing the Fields

We may pick or unselect fields from the list of all fields to determine which ones will be displayed. When you click on all fields, a window appears with a list of all the fields. Some of these fields have check marks next to them, indicating that they have already been chosen. We may utilize the check boxes to select which fields to show.

Aside from the field's name, it shows the number of different values it has, the data type it uses, and the proportion of events it appears in.

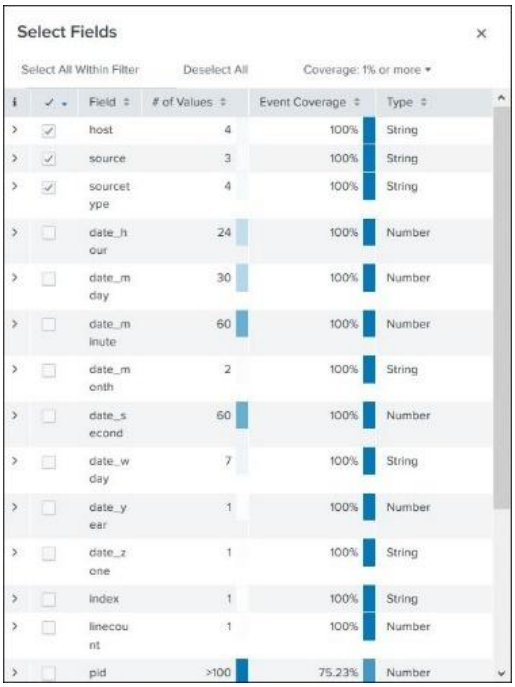


Figure 21 Choosing the Fields

Field Summary

By clicking on the name of the field, you may get more specific stats about that field. It displays all of the field's different values, as well as their counts and percentages as shown in Figure 22.

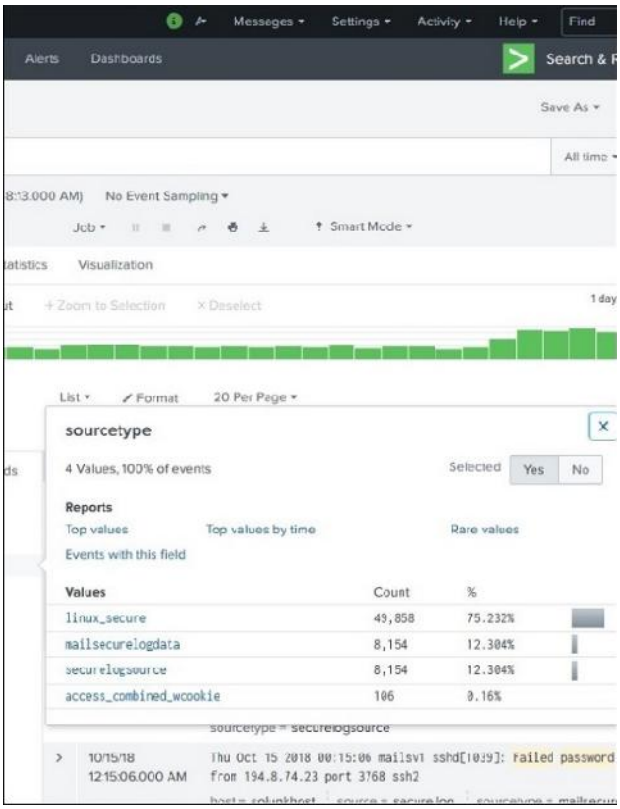


Figure 22 Field Summary

Using Fields in Search

The field names, as well as the precise values for the search, can be entered into the search box. In the example below, we're looking for all entries for the date of October 15th for the host mailsecure log. We got the outcome on this particular day.

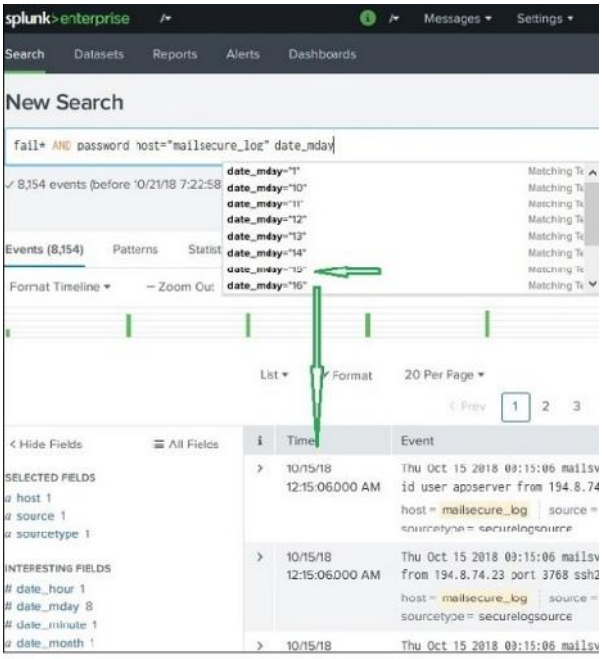


Figure 23 Using Fields in Search

14.7 DataMeer

Datameeractsas a job compiler or code generator likeHive.This means every function, filter or join that the user designs in the spreadsheet will be translated into native Tez code. Tez is great for splitting up workloads into smaller pieces. To do so, Datameer compiles a job for a Hadoop cluster, where it is sent to be executed. After the job is compiled and sent to the cluster Datameer does not control job execution, and can only receive the telemetry metrics provided by the cluster's services. The job will run with any scheduling settings and use resources granted by the scheduler.All users working with Datameer's Excel-like User Interface (UI) are generating a Java program for distributed computing on the cluster backend. This high level of abstraction is one of the key features that makes Datameer such an outstanding technology. However, this approach does mean that business users need to keep in mind the types of problems every programmer deal with, i.e., data types, memory, and disk usage.These separates analytics work into two stages. First, the design/edit time and second the execution/runtime of a data link/import job/workbook. Both stages are located on different parts within your distributed computing system (cluster).

DESIGN/EDIT TIME

The first stage is served on the Datameer application server, running the Datameer service Java Virtual Machine (JVM), started and executed under the Datameer serviceaccount user. Depending on your configuration and if (Secure) Impersonation is configured or not, calls are made from <datameerServiceAccountUser> @ <datameerHost> or <loggedinUser> @ <datameerHost>.

EXECUTION/RUN TIME

The second stage is served on random DataNodes (DN) in the cluster. The DN is running the container JVM, started by the ApplicationMaster (AM) and executed under the YARN service account user. Depending on the configuration and if (Secure) Impersonation is configured or not, callsaremade
from <yarnServiceAccountUser>@<dataNode> or <impersonatedUser>@<dataNode>

14.8 What is Data Preparation and Feature Engineering inDatameer

Data preparation is the process of cleaning, structuring, and enriching raw data, including unstructured or big data. The results are consumable data assets used for business analysis projects.In the data science community, data preparation is often called feature engineering. Although data prep and feature engineering are used interchangeably, feature engineering relies on domain-specific knowledge compared to the standard data prep process. Feature engineering creates "features" for specific machine learning algorithms, while data prep is used to disseminate data for mass consumption.Both data preparation and feature engineering are the most time-consuming and vital processes in data mining. Having data prepared correctly improves the accuracy of the outcomes. However, data preparation activities tend to be routine, tedious, and time-consuming.Datameer supports all the critical aspects of data preparation, including:

- ***Data cleansing*** – functions for the removal of bad records, replacing invalid or blank values, and de-duplicating data,
- ***Data blending*** – join and union functions to blend disparate datasets into a common, normalized view, Advanced transformations – pivoting, encoding, date and time, conversion, working with lists, parsing functions,
- ***Data enrichment*** – functions to create value-added columns including math, statistical, trigonometric, mining, and path construction,
- ***Data grouping and organization*** – more sophisticated ways to group, aggregate, and slide-and-dice data, including pivot tables, sessionization, custom binning, time windows, statistical grouping, and algorithmic grouping,
- ***Data science-specific*** – one-hot, date/time, and binned encoding functions for data science models.

Datameer can provide the universal tool for all your data transformation needs, whether data engineering, analytics engineering, and analyst or data scientist data preparation, and facilitatecataloging and collaboration across all these functions.

14.9 Installing Splunk Enterprise on Windows

You can install Splunk Enterprise on Windows with the Graphical User Interface (GUI)-based installer or from the command line. More options, such as silent installation, are available if you install from the command line. You cannot install or run the 32-bit version of Splunk Enterprise for Windows on a 64-bit Windows machine. You also cannot install Splunk Enterprise on a machine that runs an unsupported OS. For example, you cannot install Splunk Enterprise on a machine that runs Windows Server 2003. See System requirements. If you attempt to run the installer in such a way, it warns you and prevents the installation.

Before you install

- **Choose the Windows user Splunk should run as**

Before installing, see Choose the Windows user Splunk should run as to determine which user account Splunk should run as to address your specific needs.The user you choose has ramifications on what you must do prior to installing the software,

- **Disable or limit antivirus software if able**

The indexing subsystem in Splunk Enterprise demands a lot of disc traffic. Any programmer that acts as a middleman between Splunk Enterprise and the operating system might limit the amount of processing power accessible to Splunk Enterprise, resulting in slowness or even unresponsiveness. Anti-virus software is included in this category.Before you begin a Splunk installation, you must setup such software to prevent on-access scanning of Splunk Enterprise installation directories and processes.

- **Consider installing Splunk software into a directory with a short path name**

Splunk MSI files install the programme to Program FilesSplunk on the system disc by default (the drive that booted your Windows machine.) While this directory is OK for most Splunk software installations, it may be an issue for installations that are dispersed or use sophisticated. Splunk capabilities like search-head or indexer clustering. The MAX PATH path restriction in the Windows API is 260 characters long, including the drive letter, colon, backslash, 256-character path, and a null termination character, according to Microsoft. Windows cannot handle a file path greater than this, and if Splunk software creates a file with a path length longer than MAX PATH, it will be unable to access it later. There is no method to modify the settings.If you know the instance will be part of a search head or indexer cluster, try putting the pro gramme in a directory with a short path length, such as C:Splunk or D:SPL, to avoid this problem.

Install Splunk Enterprise via the GUI installer

The Windows installer is an MSI file.

Begin the installation

- From the Splunk download page, get the Splunk installer.
- To continue the installation, check the "Check this box to accept the License Agreement" checkbox. This activates the "Customize Installation" and "Next" buttons. (Optional) If you want to view the license agreement, click **View License Agreement**.

Installation Options

- You have two options when installing Windows: use the default installation settings or change all settings before installing.
- The installer does the following when you opt to install with the default settings:

  Splunk Enterprise is installed under Program Files Splunk on the disc from which your Windows computer booted.

  Splunk Enterprise is installed using the default administration and Web network ports.

- Splunk Enterprise is configured to operate as the Local System user.

  It asks you to establish a password for the Splunk administrator.

- This must be completed before the installation may proceed.
  Creates a shortcut to the software on the Start Menu.
- If you want to change any of these default installation settings, click **Customize Options** and proceed with the instructions in "Customize Options" in this topic.
- Otherwise, click **Next**. You will be prompted for a password for the Splunk admin user. After you supply a password, installation begins and you can continue with the "Complete the install" instructions.

Customize options during the installation

Several settings can be customised throughout the installation process. The installer displays the "Install Splunk Enterprise to" screen when you select to modify settings. Splunk Enterprise is installed by default under Program FilesSplunk on the system disc.The Splunk Enterprise installation directory is referred to as $SPLUNK HOME or percent SPL UNK HOME percent throughout this documentation set.Splunk Enterprise instals and operates the splunkd and splunkweb Windows services. Splunk Enterprise operations are handled by the splun kd service, whereas the splunkweb service is solely installed to run in legacy mode.The user you ch oose on the "Choose the user Splunk Enterprise should run as" screen instals and runs these service s. Splunk Enterprise may be launched as the Local System user or as another user.

- When the installer asks you the user that you want to install Splunk Enterprise as, you must specify the user name in domain\username format. The user must be a valid user in your security context, and must be an active member of an Active Directory domain.
- Splunk Enterprise must run under either the Local System account or a valid user account with a valid password and local administrator privileges. Failure to include the domain name with the user will cause the installation to fail.
- Click Change… to specify a different location to install Splunk Enterprise, or click Next to accept the default value. The installer displays the "Choose the user Splunk Enterprise should run as" panel.
- Select a user type and click **Next**.
- If you selected the Local System user, proceed to Step 5. Otherwise, the installer displays the **Logon Information: specify a username and password** panel.
- Enter the Windows credentials that Splunk Enterprise uses to run on the machine and click **Next**.
- These credentials are different from the Splunk administrator credentials that you create in the next step.
- Create credentials for the Splunk administrator user by entering a username and password that meets the minimum eligibility requirements as shown in the panel and click **Next**.
- You must perform this action as the installation cannot proceed without your completing it. If you do not enter a username, the installer creates the admin user during the installation process.
- The installer displays the installation summary panel.
- Click "Install" to proceed with the installation.
- **Complete the installation**
- The installer runs, installs the software, and displays the **Installation Complete** panel.
- If you specified the wrong user during the installation procedure, you will see two pop-up error windows explaining this. If this occurs, Splunk Enterprise installs itself as the Local System user by default. Splunk Enterprise does not start automatically in this situation.
- You can proceed through the final panel of the installation, but uncheck the "Launch browser with Splunk" checkbox to prevent your browser from launching. Then, use theseinstructions to switch to the correct user before starting Splunk.

- (Optional) Check the boxes to Launch browser with Splunk and Create Start Menu Shortcut. Click Finish. The installation completes, Splunk Enterprise starts and launches in a supported browser if you checked the appropriate box.

Install or upgrade license

- If this is a new installation of Splunk Enterprise or switching from one license type to another, you must install or update your license.
- You cannot install or run the 32-bit version of Splunk Enterprise for Windows on a 64-bit Windows machine. You also cannot install Splunk Enterprise on a machine that runs an unsupported OS. For example, you cannot install Splunk Enterprise on a machine that runs Windows Server 2003. See System requirements. If you attempt to run the installer in such a way, it warns you and prevents the installation.

Summary

- Splunk is a tool for tracking and searching large amounts of data. It indexes and correlates data in a searchable container and allows for the generation of alerts, reports, and visualisations.
- Splunk enterprise's goal is to help you figure out what's going on in your company and take action swiftly.
- Splunk cloud is a versatile, secure, and cost-effective data platform service that allows you to search, analyse, visualise, and act on your data.
- Splunk Light solves this problem by allowing you to collect and correlate data from almost any source, format, or location. Data flowing from packaged and client applications, app servers, web servers, databases, network wire data, virtual machines, operating systems, sensors, and other sources are just a few of the possibilities.
- The process of acquiring and importing data for immediate use or storage in a database is known as data intake. Ingesting anything means "to take in or absorb something." Data can be ingested in batches or broadcast in real time.
- Indexing is a technique for improving database speed by reducing the number of disc accesses necessary when a query is run. It's a data structure strategy for finding and accessing data in a database rapidly. A few database columns are used to generate indexes.
- Panel-based displays are known as dashboards. Modules like as search boxes, fields, charts, tables, and lists can be included in the panels. Reports are frequently linked to dashboard panels. You may add a search visualisation or a report to a new or existing dashboard after you build it.
- The structure of your data is defined by a Splunk data model, which is a hierarchy of datasets. Your data model should represent the data's basic structure as well as the Pivot reports that your end users demand.

Keywords

Splunk: Splunk is a search and analysis tool for machine data. Machine data might originate from online applications, sensors, devices, or any data that the user has developed. It supports IT infrastructure by analysing logs created during various operations, but it may also evaluate any organised or semi-structured data with correct data modelling.

Splunk Interface: Splunk's web interface includes all of the tools you'll need to search, report, and analyse the data you've ingested. The same web interface allows administrators to manage users

and their responsibilities. It also includes connections for data intake as well as Splunk's built-in applications.

Datameer bills itself as an all-in-one analytics solution. They help ingest (in Hadoop), cleanse/prepare data that has been ingested or is being ingested, and then query data using Hive/Tex/Spark, as well as give visualisation for the searched data, according to Datameer.

Data Ingestion: Splunk accepts a wide range of data types, including JSON, XML, and unstructured machine data such as web and application logs. The user can model the unstructured data into a data structure as desired.

Data Indexing: Splunk indexes the imported data for quicker searching and querying under various situations.

Data Searching: In Splunk, searching entails utilising the indexed data to create metrics, forecast future trends, and spot patterns.

Using Alerts: When certain criteria are identified in the data being examined, Splunk alerts may be used to send emails or RSS feeds.

Dashboards: Splunk Dashboards may display search results as charts, reports, and pivot tables, among other things.

Data Model: Based on specialized domain knowledge, the indexed data can be modelled into one or more data sets. This makes it easy for end users to navigate and evaluate business cases without having to grasp the intricacies of Splunk's search processing language.

Hadoop: Hadoop is an open-source software framework for storing and processing data on commodity hardware clusters. It has a lot of storage for any sort of data, a lot of processing power, and it can perform almost unlimited concurrent processes or jobs.

Application Master: The Application Master is a framework-specific library that is in charge of negotiating resources with the Resource Manager and working with the Node Manager(s) to execute and monitor Containers and their resource usage. It is in charge of negotiating suitable resource Containers with the Resource Manager and keeping track of their progress. The Resource Manager monitors the Application Master, which operates as a single Container.

NameNode is a component of the Master System. Namenode's main function is to manage all of the MetaData. The list of files saved in HDFS is known as metadata (Hadoop Distributed File System). In a Hadoop cluster, data is stored in the form of blocks, as we all know.

Self Assessment

1. Splunk is a software used to _____ machine data.

A. search and attention

B. search and analyze

C. surfing and analyze

D. none of the mentioned

2. The Administrator drop down menu allows you to customize and modify the _____ information

A. Administrator's

B. Reporting

C. Customer

D. User

3. The link to _____ brings us to the features where we can locate the data sets that are accessible for searching the reports and alerts that have been produced for these searches.

A. search and records
B. short and reporting
C. search and reporting
D. None of the above

4. Data ingestion in Splunk happens through the _____ feature which is part of the search and reporting app.

A. Add data
B. Upload data
C. Ingest data
D. None of the above

5. What is called the process of transforming the data?

A. Events
B. Repository
C. Indexing
D. None of the above

6. Which option to use to review the input settings.

A. Source
B. Review
C. Both
D. None of the above

7. Use of TEZ is to _____

A. Splitting up workloads into smaller pieces.
B. Uploading the data
C. Adding the data
D. None of the above

8. Which of the following problems every programmer deals and business users need to keep in mind?

A. Datatype
B. Memory
C. Disk usage
D. All of the above

9. Select the stages in which analytics work.

A. Design/edit time
B. Execution/run time

C. Both
D. None of the above

10. Select the options that can be used to install splunk enterprise on windows
a. GUI interface
b. Command Line Interface
c. Both
d. None of the above

11. Select the parameter(s) that prevent users from installing Splunk.
A. Unsupported OS
B. Windows Server 2003
C. Both
D. None of the above

12. The MAX PATH path restriction in the Windows API is ___ characters long.
A. 250
B. 260
C. 270
D. None of the above

13. Which feature of Splunk is used to search the entire data set that is ingested.
A. Search & Reporting
B. Refining search results
C. Using fields in search
D. Sharing the search result.

14. What are the different formats are available for exports?
A. CSV
B. XML
C. JSON
D. All of the above

15. Which of the following are the components of the SPLUNK search processing language (SPL)?
A. Search terms
B. Commands
C. Functions
D. All of the above

Answers for Self Assessment

1.	B	2.	A	3.	C	4.	B	5.	C
6.	B	7.	A	8.	D	9.	C	10.	C
11.	C	12.	B	13.	A	14.	D	15.	D

Review Questions

1) Write down steps forinstalling splunk enterprise on windows
2) What is data preparation and datameer?
3) Write down functions of search and reporting app?
4) What are the different types of Splunk dashboards and also write down components of Splunk architecture?
5) What are the benefits of feeding data into a Splunk instance through Splunk Forwarders?

Further Readings

- Maheshwari, Anil. *Big Data*. McGraw-Hill Education, 2019.
- Mayer-Schonberger, Viktor; Cukier, Kenneth (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think . Houghton Mifflin Harcourt.
- McKinsey Global Institute Report (2011). Big Data: The Next Frontier For Innovation, Competition, and Productivity. Mckinsey.com
- Marz, Nathan, and James Warren (2015). Big Data: Principles and Best Practices of Scalable Real time Data Systems. Manning Publications.
- Sandy Ryza, Uri Laserson et.al (2014). Advanced-Analytics-with-Spark. OReilley.
- White, Tom (2014). Mastering Hadoop. OReilley.

Web Links

1. Apache Hadoop resources: https://hadoop.apache.org/docs/r2.7.2/
2. Apache HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
3. Hadoop API site: http://hadoop.apache.org/docs/current/api/
4. NOSQL databases: http://nosql-database.org/
5. Apache Spark: http://spark.apache.org/docs/latest/
6. Tutorials on Big Data technologies: https://www.tutorialspoint.com/

LOVELY PROFESSIONAL UNIVERSITY
Jalandhar-Delhi G.T. Road (NH-1)
Phagwara, Punjab (India)-144411
For Enquiry: +91-1824-521360
Fax.: +91-1824-506111
Email: odl@lpu.co.in