H1B VISA PROJECT

BY
BETTINA JEBARAJ
S181165200220

H1-B PROJECT

INTRODUCTION:

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. In this project we will be performing analysis on the H1B visa applicants between the years 2011-2016. The H1B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign national to apply for H1B visa, an US employer must offer a job and petition for H1B visa with the US immigration department.

BIGDATA:

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 involves many areas of business and technology.

What Comes Under Big Data?

Big data involves the data produced by different devices and applications. Given below are some of the fields that come under the umbrella of 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**: 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.

Five Vs of Big Data:

- Volume :Scale of data
- Velocity :Analysis of streaming data
- Variety : Different forms of data
- Veracity :Uncertainty of data

Thus Big Data includes huge volume, high velocity, and extensible variety of data. The data in it will be of three types.

- Structured data: Relational data.
- Semi Structured data: XML data.
- Unstructured data: Word, PDF, Text, Media Logs.

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 data regarding the previous medical history of patients, hospitals are providing better and quick service

HADOOP:

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.

Hadoop Architecture at its core, Hadoop has two major layers namely:

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

MapReduce:

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.

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.

How Does Hadoop Work?

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, you can tie together many commodity computers with single-CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. Moreover, it is cheaper than one high-end server. So this is the first motivational factor behind using Hadoop that it runs across clustered and low-cost machines.

Hadoop runs code across a cluster of computers. This process includes the following core tasks that Hadoop performs:

- Data is initially divided into directories and files. Files are divided into uniform sized blocks of 128M and 64M (preferably 128M).
- These files are then distributed across various cluster nodes for further processing.
- HDFS, being on top of the local file system, supervises the processing.
- Blocks are replicated for handling hardware failure.
- Checking that the code was executed successfully.
- Performing the sort that takes place between the map and reduce stages.
- Sending the sorted data to a certain computer.
- Writing the debugging logs for each job.

Advantages of Hadoop :

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
- Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
- Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

HADOOP ECO-SYSTEM:

The Hadoop ecosystem includes both official Apache open source projects and a wide range of commercial tools and solutions. Some of the best-known open source examples include Spark, Hive, Pig, Oozie and Sqoop. Commercial Hadoop offerings are even more diverse and include platforms and packaged distributions from vendors such as Cloudera, Hortonworks, and MapR, plus a variety of tools for specific Hadoop development, production, and maintenance tasks. Most of the solutions available in the Hadoop ecosystem are intended to supplement one or two of Hadoop's four core elements (HDFS, MapReduce, YARN, and Common). However, the commercially available framework solutions provide more comprehensive functionality.

Apache open source Hadoop ecosystem elements:

The Apache Hadoop project actively supports multiple projects intended to extend Hadoop's capabilities and make it easier to use. There are several top-level projects to create development tools as well as for managing Hadoop data flow and processing. Many commercial third-party solutions build on the technologies developed within the Apache Hadoop ecosystem.

Spark, Pig, and Hive are three of the best-known Apache Hadoop projects. Each is used to create applications to process Hadoop data. While there are a lot of articles and discussions about whether Spark, Hive or Pig is better, in practice many organizations do not only use a single one because each is optimized for specific functions.

Spark

Spark is both a programming model and a computing model. It provides a gateway to in-memory computing for Hadoop, which is a big reason for its popularity and wide adoption. Spark provides an alternative to MapReduce that enables workloads to execute in memory, instead of on disk. Spark accesses data from HDFS but bypasses the MapReduce processing framework, and thus eliminates the resource-intensive disk operations that MapReduce requires. By using in-memory computing, Spark workloads typically run between 10 and 100 times faster compared to disk execution.

Spark can be used independently of Hadoop. However, it is used most commonly with Hadoop as an alternative to MapReduce for data processing. Spark can easily coexist with MapReduce and with other ecosystem components that perform other tasks.

Spark is also popular because it supports SQL, which helps overcome a shortcoming in core Hadoop technology. The Spark programming environment works interactively with Scala, Python, and R shells. It has been used for data extract/transform/load (ETL) operations, stream processing, machine learning development and with the Apache GraphX API for graph computation and display. Spark can run on a variety of Hadoop and non-Hadoop clusters, including Amazon S3.

Hive

Hive is data warehousing software that addresses how data is structured and queried in distributed Hadoop clusters. Hive is also a popular development environment that is used to write queries for data in the Hadoop environment. It provides tools for ETL operations and brings some SQL-like capabilities to the environment. Hive is a declarative language that is used to develop applications for the Hadoop environment, however it does not support real-time queries.

Hive has several components, including:

- HCatalog – Helps data processing tools read and write data on the grid. It supports MapReduce and Pig.
- WebHCat – Lets you use an HTTP/REST interface to run MapReduce, Yarn, Pig, and Hive jobs.
- HiveQL – Hive's query language intended as a way for SQL developers to easily work in Hadoop. It is similar to SQL and helps both structure and query data in distributed Hadoop clusters.

Hive queries can run from the Hive shell, JDBC, or ODBC. MapReduce (or an alternative) breaks down HiveQL statements for execution across the cluster.

Hive also allows MapReduce-compatible mapping and reduction software to perform more sophisticated functions. However, Hive does not allow row-level updates or support for real-time queries, and it is not intended for OLTP workloads. Many consider Hive to be much more effective for processing structured data than unstructured data, for which Pig is considered advantageous.

Pig

Pig is a procedural language for developing parallel processing applications for large data sets in the Hadoop environment. Pig is an alternative to Java programming for MapReduce, and automatically generates MapReduce functions. Pig includes Pig Latin, which is a scripting language. Pig translates Pig Latin scripts into MapReduce, which can then run on YARN and process data in the HDFS cluster. Pig is popular because it automates some of the complexity in MapReduce development.

Pig is commonly used for complex use cases that require multiple data operations. It is more of a processing language than a query language. Pig helps develop applications that aggregate and sort data and supports multiple inputs and exports. It is highly customizable, because users can write their own functions using their preferred scripting language. Ruby, Python and even Java are all supported. Thus, Pig has been a popular option for developers that are familiar with those languages but not with MapReduce. However, SQL developers may find Hive easier to learn.

HBase

HBase is a scalable, distributed, NoSQL database that sits atop the HFDS. It was designed to store structured data in tables that could have billions of rows and millions of columns. It has been deployed to power historical searches

through large data sets, especially when the desired data is contained within a large amount of unimportant or irrelevant data (also known as sparse data sets). It is also an underlying technology behind several large messaging applications, including Facebook's.

HBase is not a relational database and wasn't designed to support transactional and other real-time applications. It is accessible through a Java API and has ODBC and JDBC drivers. HBase does not support SQL queries, however there are several SQL support tools available from the Apache project and from software vendors. For example, Hive can be used to run SQL-like queries in HBase.

Oozie

Oozie is the workflow scheduler that was developed as part of the Apache Hadoop project. It manages how workflows start and execute, and also controls the execution path. Oozie is a server-based Java web application that uses workflow definitions written in hPDL, which is an XML Process Definition Language .Oozie only supports specific workflow types, so other workload schedulers are commonly used instead of or in addition to Oozie in Hadoop environments.

Sqoop

Think of Sqoop as a front-end loader for big data. Sqoop is a command-line interface that facilitates moving bulk data from Hadoop into relational databases and other structured data stores. Using Sqoop replaces the need to develop scripts to export and import data. One common use case is to move data from an enterprise data warehouse to a Hadoop cluster for ETL processing. Performing ETL on the commodity Hadoop cluster is resource efficient, while Sqoop provides a practical transfer method.

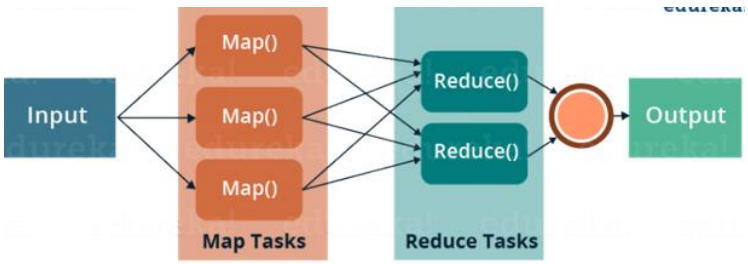
TOOLS USED IN THE PROJECT:

MapReduce:

What is MapReduce?

MapReduce is a programming framework that allows us to perform distributed and parallel processing on large data sets in a distributed environment.

- MapReduce consists of two distinct tasks – Map and Reduce.
- As the name MapReduce suggests, reducer phase takes place after mapper phase has been completed.
- So, the first is the map job, where a block of data is read and processed to produce key-value pairs as intermediate outputs.
- The output of a Mapper or map job (key-value pairs) is input to the Reducer.
- The reducer receives the key-value pair from multiple map jobs.
- Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.



Advantages of MapReduce**:**

The two biggest advantages of MapReduce are:

1. Parallel Processing:

In MapReduce, we are dividing the job among multiple nodes and each node works with a part of the job simultaneously. So, MapReduce is based on Divide and Conquer paradigm which helps us to process the data using different machines. As the data is processed by multiple machine instead of a single machine in parallel, the time taken to process the data gets reduced by a tremendous amount.

2. Data Locality:

Instead of moving data to the processing unit, we are moving processing unit to the data in the MapReduce Framework. In the traditional system, we used to bring data to the processing unit and process it. But, as the data grew and became very huge, bringing this huge amount of data to the processing unit posed following issues:

- Moving huge data to processing is costly and deteriorates the network performance.
- Processing takes time as the data is processed by a single unit which becomes the bottleneck.
- Master node can get over-burdened and may fail.

Now, MapReduce allows us to overcome above issues by bringing the processing unit to the data. The data is distributed among multiple nodes where each node processes the part of the data residing on it. This allows us to have the following advantages:

- It is very cost effective to move processing unit to the data.
- The processing time is reduced as all the nodes are working with their part of the data in parallel.
- Every node gets a part of the data to process and therefore, there is no chance of a node getting overburdened.

Pig:

What is Apache Pig?

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. 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.

Features of Pig

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.

Applications of Apache Pig

Apache Pig is generally used by data scientists for performing tasks involving ad-hoc processing and quick prototyping. Apache Pig is used −

- To process huge data sources such as web logs.
- To perform data processing for search platforms.
- To process time sensitive data loads.

Hive:

What is Hive?

Hive is an ETL and Data warehousing tool developed on top of Hadoop Distributed File System (HDFS). Hive makes job easy for performing operations like

- Data encapsulation
- Ad-hoc queries
- Analysis of huge datasets

Important characteristics of Hive

- In Hive, tables and databases are created first and then data is loaded into these tables.
- Hive as data warehouse designed for managing and querying only structured data that is stored in tables.
- While dealing with structured data, Map Reduce doesn't have optimization and usability features like UDFs but Hive framework does. Query optimization refers to an effective way of query execution in terms of performance.
- Hive's SQL-inspired language separates the user from the complexity of Map Reduce programming. It reuses familiar concepts from the relational database world, such as tables, rows, columns and schema, etc. for ease of learning.
- Hadoop's programming works on flat files. So, Hive can use directory structures to "partition" data to improve performance on certain queries.

- A new and important component of Hive i.e. Metastore used for storing schema information. This Metastore typically resides in a relational database. We can interact with Hive using methods like
    - Web GUI
    - Java Database Connectivity (JDBC) interface
- Most interactions tend to take place over a command line interface (CLI). Hive provides a CLI to write Hive queries using Hive Query Language(HQL)
- Generally, HQL syntax is similar to the SQL syntax that most data analysts are familiar with. The Sample query below display all the records present in mentioned table name.
    - Sample query : Select * from <TableName>
- Hive supports four file formats those are TEXTFILE, SEQUENCEFILE, ORC and RCFILE (Record Columnar File).
- For single user metadata storage, Hive uses derby database and for multiple user Metadata or shared Metadata case Hive uses MYSQL.

Different modes of Hive

Hive can operate in two modes depending on the size of data nodes in Hadoop.

These modes are,

- Local mode
- Map reduce mode

When to use Local mode:

- If the Hadoop installed under pseudo mode with having one data node we use Hive in this mode
- If the data size is smaller in term of limited to single local machine, we can use this mode
- Processing will be very fast on smaller data sets present in the local machine

When to use Map reduce mode:

- If Hadoop is having multiple data nodes and data is distributed across different node we use Hive in this mode
- It will perform on large amount of data sets and query going to execute in parallel way
- Processing of large data sets with better performance can be achieved through this mode

Sqoop:

What is Sqoop?

Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and external datastores such as relational databases, enterprise data warehouses.

Sqoop is used to import data from external datastores into Hadoop Distributed File System or related Hadoop eco-systems like Hive and HBase. Similarly, Sqoop can also be used to extract data from Hadoop or its eco-systems and

export it to external datastores such as relational databases, enterprise data warehouses. Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres etc.

Where is Sqoop used?

Relational database systems are widely used to interact with the traditional business applications. So, relational database systems has become one of the sources that generate Big Data.As we are dealing with Big Data, Hadoop stores and processes the Big Data using different processing frameworks like MapReduce, Hive, HBase, Cassandra, Pig etc and storage frameworks like HDFS to achieve benefit of distributed computing and distributed storage. In order to store and analyze the Big Data from relational databases, Data need to be transferred between database systems and Hadoop Distributed File System (HDFS). Here, Sqoop comes into picture. Sqoop acts like a intermediate layer between Hadoop and relational database systems. You can import data and export data between relational database systems and Hadoop and its eco-systems directly using sqoop. Sqoop provides command line interface to the end users. Sqoop can also be accessed using Java APIs. Sqoop command submitted by the end user is parsed by Sqoop and launches Hadoop Map only job to import or export data because Reduce phase is required only when aggregations are needed. Sqoop just imports and exports the data; it does not do any aggregations.

Sqoop parses the arguments provided in the command line and prepares the Map job. Map job launch multiple mappers depends on the number defined by user in the command line. For Sqoop import, each mapper task will be assigned with part of data to be imported based on key defined in the command line. Sqoop distributes the input data among the mappers equally to get high performance. Then each mapper creates connection with the database using JDBC and fetches the part of data assigned by Sqoop and writes it into HDFS or Hive or HBase based on the option provided in the command line.

CONCLUSION:

Thus the H1B Visa data has been analysed with queries and coding using tools like mapreduce, pig, hive and scoop and corresponding results have been obtained and saved.