MINI PROJECT - II

(2020-21)

Country Data Analysis Using Hadoop

SYNOPSIS

Institute of Engineering &Technology

SUBMITTED BY

Shimanshu Sharma

(181599003)

Supervised By

Mr. Harvinder Kaur
(Technical Trainer)

Department of Computer Engineering & Applications

About the Project:

Analysis of structured data has seen tremendous success in the past. However, analysis of large scale unstructured data remains a challenging area. The main objective of this project is to demonstrate by using Hadoop concepts. The text file output generated from the console application is then loaded from HDFS.

HDFS is a primary Hadoop application and a user can directly interact with HDFS using various shell-like commands supported by Hadoop. This project uses No SQL like queries that are later run on Big Data using PIG to extract the meaningful output.

In this study we have used United Nations historical data of all countries' flag details.

We collect the data to keep in HDFS (Hadoop Distributed File Systems) and to analyze it using PIG Latin on HADOOP storage to find out various queries asked by the Data Contributor

Motivation:

1. To introduce new products and services.
2. To reduce transaction costs.
3. To meet customers demand.

Features:

The system will comprises of 4 modules as follows:

1.Project :- Data Analysis using HADOOP ECO-SYSTEM – PIG Latin

2.Title :- United Nation Flag Data Analysis

3.Tools :-Oracle VM

4.Operating System :- CentOS

5.Language :- Procedural Language - PIG LATIN

6.Framework used – Apache HADOOP 2.x

7.Data Base :- NoSql database – HDFS [Hadoop distributed File System]

8.Nature of Project: - Data Analysis

9.Data Type :- Schema less Data

TABLE OF CONTENTS

INTRODUCTION

Machine-generated data is growing exponentially from the last several years. This data is generated in social networking sites via posts from many users, sensor data: to get climate information, purchase transaction records in large industries and many more. With the help of normal legacy systems, it becomes very difficult and expensive to store and analyze large scale data for data analysts. It is also a time consuming process. This kind of large scale data with structured and unstructured format is called Big Data. However, Hadoop framework is growing now days to store and analyze data and it is convenient for its functions.Hive is one of the ecosystems in Hadoop framework which is built by Facebook to analyze the data on Hadoop clusters. Hive syntax is based on SQL, so a a person with the knowledge of SQL can easily work in a Hive environment. The syntax used in Hive is called Hive QL (Hive Query Language). Many companies have been using big data framework to analyze the data and find some patterns and relationship among the data to target customer and market competition. In this study we have used United Nations historical data of all countries' flag details. We collect the data to keep in HDFS (Hadoop Distributed File Systems) and to analyze it using Hive & PIG Latin on HADOOP storage to find out various queries asked by the Data Contributor .
Whereas RDBMS is designed to handle structured data and that to only certain limit, RDBMS fails to handle this kind of unstructured and huge amount of data called Big Data. This inability of RDBMS has given birth to new database management system called NoSQL management system.

Some of the key concepts used in Big Data Analytics are:
Data Mining: Data mining is incorporation of quantitative methods. Using powerful mathematical techniques applied to analyze data and how to process that data. It is used to extract data and find actionable information which is used to increase productivity and efficiency**.**

Overview of the Project

BIG DATA AND HADOOP

What is Big Data?
Big Data is a collection of data sets so large and complex that it becomes difficult to

process using the available database management tools. The challenges include how to capture, curate, store, search, share, analyze and visualize Big Data. In today's environment, we have access to more types of data. These data sources include online transactions, social networking activities, mobile device services, internet gaming etc.

Big Data is a collection of data sets that are large and complex in nature. They constitute both structured and unstructured data that grow large so fast that they are not manageable by traditional relational database systems or conventional statistical tools.

Big Data is defined as any kind of data source that has at least three shared characteristics:

Extremely large Volumes of data

Extremely high Velocity of data

Extremely wide Variety of data

According to Big Data concepts, methodologies, tools and applications, volume by Information Resources Management Association organizations today are at the tipping point in terms of managing data. Data sources are ever expanding. Data from Facebook, Twitter, YouTube, Google etc., are to grow 50X in the next 10 years. Over 2.5 exabytes of data is generated every day. Some of the sources of huge volume of data are:

A typical large stock exchange captures more than 1 TB of data everyday.There are over 5 billion mobile phones in the world which are producing enormous amount of data on daily basis.YouTube users upload more than 48 hours of video every minute.

Large social networks such as Twitter and Facebook capture more than 10 TB of data daily.There are more than 30 million networked sensors in the world which further produces TBs of data every day.

Structured and semi-structured formats have some limitations with respect to handling large quantities of data. Hence, in order to manage the data in the Big Data world, new emerging approaches are required, including document, graph, columnar, and geospatial database architectures. Collectively, these are referred to as NoSQL, or not only SQL, databases. In essence the data architectures need to be mapped to the types of transactions. Doing so will help to ensure the right data is available when you need it.

What is Hadoop?

As organizations are getting flooded with massive amount of raw data, the challenge here is that traditional tools are poorly equipped to deal with the scale and complexity of such kind of data. That 's where Hadoop comes in. Hadoop is well suited to meet many Big Data challenges, especially with high volumes of data and data with a variety of structures.

At its core, Hadoop is a framework for storing data on large clusters of commodity hardware everyday computer hardware that is affordable and easily available and running applications against that data. A cluster is a group of interconnected computers (known as nodes) that can work together on the same problem. Using networks of affordable compute resources to acquire business insight is the key value proposition of Hadoop.

Hadoop consists of two main components:

A distributed processing framework named MapReduce (which is now supported by a component called YARN(Yet Another Resource Negotiator) and

A distributed file system known as the Hadoop Distributed File System, or HDFS. In Hadoop you can do any kind any kind of aggregation of data whether it is one month old data or one-year-old data. Hadoop provides a mechanism called MapReduce model to do distributed processing of large data which internally takes care of data even if one machine goes down.

Hadoop Ecosystem

Hadoop is a shared system where each node acts independently throughout the system. A framework where a piece of work is divided among several parallel MapReduce tasks. Each task operated independently on cheap commodity servers. This enables businesses to generate values from data that was previously considered too expensive to be stored and processed in a traditional data warehouse or Online Transaction Processing

environment. In the old paradigm, companies would use a traditional enterprise data warehouse system and would buy the biggest data warehouse they could afford and store the data on a single machine. However, with the increasing amount of data, this approach is no longer affordable nor practical.
Some of the components of Hadoop ecosystem are HDFS,MapReduce,Yarn,Hive and Hbase.Hadoop has two core components. 'Storage' part to store the data and 'Processing' part to process the data. The storage part is called 'HDFS' and the The processing part is called 'YARN'.

HDFS Architecture

Storage Component: Hadoop Distributed File System(HDFS)
HDFS is the storage component of the core Hadoop Infrastructure. HDFS provides a distributed architecture for extremely large scale storage, which can easily be extended by scaling out. It is important to mention the difference between scale up and scale out. In its initial days, Google was facing challenges to store and process not only all the pages on the internet but also its users' web log data. At that time, Google was using scale up architecture model where you can increase the system capacity by adding CPU

cores, RAM etc to the existing server. But such kind of model had was not only expensive but also had structural limitations. So instead, Google engineers implemented Scale out architecture model by using cluster of smaller servers which can be further scaled out if they require more power and capacity. Google File System (GFS) was developed based on this architectural model. HDFS is designed based on similar concept.
The core concept of HDFS is that it can be made up of dozens, hundreds, or even thousands of individual computers, where the system#39;s files are stored in directly attached disk drives. Each of these individual computers is a self-contained server with its own memory, CPU, disk storage, and installed operating system (typically Linux, though Windows is also supported). Technically speaking, HDFS is a user-space-level file system because it lives on top of the file systems that are installed on all individual computers that make up the Hadoop cluster.
Hadoop cluster is made up of two classes of servers: slave nodes, where the data is stored

and processed and master nodes, which govern the management of the Hadoop cluster. On each of the master nodes and slave nodes, HDFS runs special services and stores raw data to capture the state of the file system. In the case of the slave nodes, the raw data consists of the blocks stored on the node, and with the master nodes, the raw data consists of metadata that maps data blocks to the files stored in HDFS.

HDFS is a system that allows multiple commodity machines to store data from a single source. HDFS consists of a NameNode and a DataNode. HDFS operates as master slave architecture as opposed to peer to peer architecture. NameNode serves as the master component while the DataNode serves as a slave component. NameNode comprises of only the Metadata information of HDFS that is the blocks of data that are present on the Data Node.

The DataNode comprises of data processing, all the processing data that is stored on the DataNode and deployed on each machine.

The actual storage of the files being processed and serving read and write request for the clients In the earlier versions of Hadoop there was only one NameNode attached to the DataNode which was a single point of failure. Hadoop version 2.x provides multiple NameNode where secondary NameNode can take over in the event of a

primary.NameNode failure. Secondary NameNode is responsible for performing periodic check points in the event of a primary NameNode failure. You can start secondary NameNode by providing checkpoints that provide high availability within HDFS.

Let’s take look at a data warehouse structure example where we have one machine and with HDFS we can distribute the data into more than one machine. Let’s say we have 100 GB of file that takes 20 minutes to process on a machine with a given number of channel and hard drive. If you add four machines of exactly the same configuration on a Hadoop cluster, the processing time reduces to approximately one fourth of the original processing time or about 5minutes.

But what happens if one of these four machines fails? HDFS creates a self-healing architecture by replicating the same data across multiple nodes. So it can process the data in a high availability environment. For example, if we have three DataNodes and one NameNode, the data is transferred from the client environment into HDFS DataNode.

The replication factor defines the number of times a data block is replicated in a clustered environment. Let's say we have a file that is split into two data blocks across three DataNodes. If we are processing these files to a three DataNode cluster and we set the replication factor to three. If one of the nodes fails, the data from the failed nodes is redistributed among the remaining active nodes and the other nodes will complete the processing function.

Processing Component: Yet Another Resource Negotiator(YARN)

YARN is a resource manager that identifies on which machine a particular task is going to be executed. The actual processing of the task or program will be done by Node Manager. In Hadoop 2.2, YARN augments the MapReduce platform and serves as the Hadoop operating system. Hadoop 2.2 separates the resource management function from data processing allowing greater flexibility. This way MapReduce only performs data processing while resource management is isolated in YARN. Being the primary resource manager in HDFS, YARN enables enterprises to store data in a single place and interact with it in multiple ways with consistent levels of service. In Hadoop 1.0 the

NameNode used a job tracker and the DataNode used a task tracker to manage resources. In Hadoop 2.x, YARN splits up into two major functionalities of the job tracker - the resource management and job scheduling. The client reports to the resource manager and the resource manager allocates resources to jobs using the resource container, Node Manager and app master. The resource container splits memory, CPU, network bandwidth among other hardware constraints into a single unit. The Node Manager receives updates from the resource containers which communicate with the app master. The Node Manager is the framework for containers, resource monitoring and for reporting data to the resource manager and scheduler.

Hadoop Framework

Hadoop Framework comprises of Hadoop Distributed File System and the MapReduce framework. The Hadoop framework divides the data into smaller chunks and stores divides that data into smaller chunks and stores each part of the data on a separate node within the cluster. For example, if we have 4 terabytes of data, the HDFS divides this data

into 4 parts of 1TB each. By doing this, the time taken to store the data onto the disk is significantly reduced. The total time to store this entire data onto the disk is equal to storing 1 part of the data as it will store all the parts of the data simultaneously on different machines.

In order to provide high availability what Hadoop does is replicate each part of the data onto other machines that are present within the cluster. The number of copies it will replicate depends on the replication factor. By default the replication factor is 3, in such in this case there will be 3 copies of each part of the data on three different machines. In order to reduce the bandwidth and latency time, it will store two copies on the same rack and third copy on a different rack. For example, in the above example, NODE 1 and NODE 2 are on rack one and NODE 3 and N ODE 4 are on rack two. Then the first two copies of part 1 will be stored on NODE 1 and third copy will be stored either on NODE 3 or NODE 4. Similar process is followed in storing remaining parts of the data. The HDFS takes care of the networking required by these nodes in order to communicate.

COUNTRY DATA ANALYSIS USING HADOOP

Hadoop Data Analysis Technologies

While Hadoop provides the ability to collect data on HDFS,there are many applications available in the market like MapReduce, Pig and Hive that can be used to analyze the data.

Let us first take a closer look at all three applications and then analyze which application is better suited for Flag Data Analysis projects.

MapReduce

MapReduce is a set of Java classes run on YARN with the purpose of processing massive amounts of data and reducing this data into output files. HDFS works with MapReduce to divide the data in parallel fashion on local or parallel machines. Parallel structure requires that the data is immutable and cannot be updated. It begins with the input files where the data is initially stored typically residing in HDFS. These input files

are then split up into input format which selects the files, defines the input splits, breaks the file into tasks and provides a place for record reader objects. The input format defines the list of tasks that makes up the map phase. The tasks are then assigned to the nodes in the system based on where the input files chunks are physically resident. The input split describes the unit of work that comprises a single map task in a MapReduce program.

The record reader loads the data and converts it into key value pairs that can be read by the Mapper. The Mapper performs the first phase of the MapReduce program. Given a key and a value the mappers export key and value pairs and send these values to the reducers. The process of moving mapped outputs to the reducers is known as shuffling. Partitions are the inputs to reduce tasks, the partitioner determines which key and value pairs will be stored and reduced. The set of intermediate keys are automatically stored before they are sent to the reduce function. A reducer instance is created for each

reduced task to create an output format. The output format governs the way objects are written, the output format provided by Hadoop writes the files to HDFS**.**

Hive

Hive provides the ability to store large amounts of data in HDFS. Hive was designed to appeal to a community comfortable with SQL. Hive uses an SQL like language known as HIVEQL. Its philosophy is that we don't need yet another scripting language. Hive supports maps and reduced transform scripts in the language of the user's choice which can be embedded with SQL. Hive is widely used in Facebook, with analyst comfortable with SQL as well as data miners programming in Python.

Supporting SQL syntax also means that it is possible to integrate with existing tools like. Hive has an ODBC JDBC driver that allows and facilitates easy queries. It also adds support for indexes which allows support for queries common in such environment. Hive is a framework for performing analytical queries. Currently Hive can be used to query data stored in HBase which is a key value store like those found in the gods of most RDBMS and the Hadoop database project uses Hive query RDBMS tier**.**

Pig

Pig comes from the language Pig Latin. Pig Latin is a procedural programming language and fits very naturally in the pipeline paradigm. When queries become complex with Most joints and filters then Pig is strongly recommended. Pig Latin allows pipeline developers to decide where to checkpoint data in the pipeline. That is storing data in between operations has the advantage of check pointing data in the pipeline. This ensures the whole pipeline does not have to be rerun in the event of a failure. Pig Latin allows users to store data at any point in the pipeline without disturbing the pipeline execution.
The advantage that Pig Latin provides is that pipelines developers decide where appropriate checkpoints are in the pipeline rather than being forced to checkpoint

wherever the schematics of SQL impose it. Pig Latin supports splits in the pipeline. Common features of data pipelines is that they are often graphics and not linear pipelines since disk's read and write scan time and intermediate results usually dominate processing of large data sets reducing the number of times data must be written to and read from disk is crucial for good performance.
Pig Latin allows developers to insert their own code almost anywhere in the data pipeline which is useful for pipeline development. This is accomplished through a user defined functions UDFS (User Defined Functions). UDFS allows user to specify how data is loaded, how data is stored and how data is processed. Streaming allows users to include executables at any point in the data flow. Pipeline also often includes user defined columns transformation functions and user defined aggregations. Pig Latin supports writing both of these types of functions in java.

Pig is a procedural data flow language. A procedural language is a step by step approach defined by the programmers. Pig requires a learning curve since the syntax is new and different from SQL. The values of variables may not be retained; instead, the query needs to rerun in order to get the values from a variable. Moreover, Pig is a scripting language that is more suitable for prototyping and rapidly developing MapReduce based jobs. So we will be using pig.

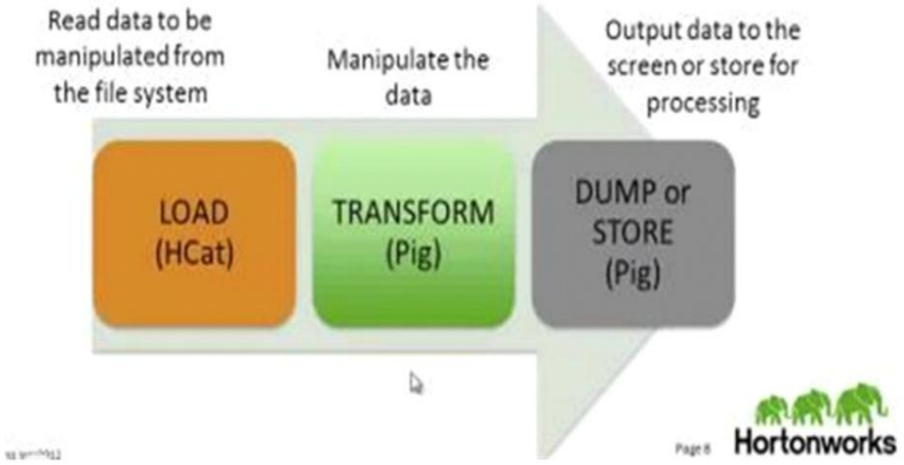
IMPLEMENTATION

Here we define the working of our project. This project is based on analysis of big data using hadoop ecosystem‘s tool pig. In this project we use tool pig. The pig tool has two modes of operation one is local mode and another is HDFS mode. In the tool pig we actually works in grunt shell and all the commands runs in grunt shell. The output also shows in grunt shell. We can store this output in the HDFS file system. There are various steps used in this project for analyzing big data. The steps are as follow:-

Used Component-

PIG

Flow diagram of PIG



Requirements and Specifications

1. **Hardware Requirements:**
    1. Processor : Intel dual Core ,i5

2. RAM : 8 GB
3. Hard disk : Minimum 1TB

2. **Software Requirements:**

Software Requirements deal with defining software resource requirements that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

1. VMWare or Virtual Box 2. Hadoop 3.Apache Pig!

Benefits

1. It supports the Stripe payment option, giving buyers the liberty to make transactions without paying any extra fees.

2. It offers more marketing tools such as custom gift cards, discount coupons, store statistics, targeted email marketing, and more.

Conclusion

The task of big data analysis is not only important but also a necessity. In Fact many organizations that have implemented Big Data are realizing significant competitive advantage compared to other organizations with no Big Data efforts. The project is intended to analyze the Flag Big Data and come up with significant insights which cannot be determined otherwise.

Flag data analysis project show key insights that can be extrapolated to other use cases as well.

BIBLIOGRAPHY

1. Big is next – Anand

https://bigishere.wordpress.com/

2. Geeks for geeks

https://www.geeksforgeeks.org/