WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Big Data Analytics (Hadoop)

Keshav Sarda

Department of Computer Science, Arya College of Engineering & I.T. Rajasthan, India

ABSTRACT:

Big Data is the latest driver of global financial and social adaptations. The world's data set is reaching a peak point for greatest technological modifications that can find new ways of outcome making, management of everyone's fitness, lands, finance and knowledge. As the intricacy of data is increasing, including the volume, velocity, variety and veracity of facts and figures, The real blow depends on our capabilities to unleash the "value" of data by Big Data analytics technologies. Big Data analytics poses a huge challenge in constructing highly scalable algos and systems to ingest the data and unleash big hidden values from raw figure sets that are differing, hard and enormous in measure. Potential break-through include the latest algorithms, methods, systems and frameworks in Big Data analytics which discover useful and buried Insights from Big Data skillfully and impactfully. Big Data analytics is related to Hong Kong in its transformation to a digital economy and people. Hong Kong has always been among the world's finest in Big Data analytics, claiming leading positions as chair and EIC of major meetings and books in Big Data-related boundaries. To maintain those leadership levels, Hong Kong's government, universities and industries must act fast to address various major complex challenges. These challenges involve "foundations," which reflect fresh algorithms, studies, and methods in knowledge hunting from massive amounts of facts and figures, and "Frameworks and applications," which include contemporary applications as well as systems helpful in encouraging Big Data activities. Big Data analytics should also be a team achievement spanning academics, government authorities, society, and industrial organizations, and by people who research from various disciplines, including data science, computer science and engineering, health, social studies and economic areas.

I. INTRODUCTION:

Apache Hadoop is a free-source software platform utilized for distributed storage as well as parallel processing of massive data sets applying the MapReduce complex programming model. It involves computing clusters created from a physical commodity that is hardware. All the Hadoop programming modules are constructed with the basic assumption that hardware downfalls are frequent occurrences and must be handled automatically via the framework.

The fundamental core of Apache Hadoop include a storage division, called as Hadoop Distributed File System (HDFS), and a computing or processing part which is called as MapReduce programming framework model. Hadoop divides files into huge blocks and then distributes them in different nodes inside a cluster. It then sends the packaged codeword to the nodes for processing the data in parallel mode. In this technique nodes manipulate the data to which they have access(data locality) to permit the information set to be processed more quickly and with more efficiency than it would have been on a more traditional supercomputer framework that count on the parallel file system where computational calculations and data set are divided across very high-speed networks.

Apache Hadoop software library is an framework that facilitates distributed processing of massive data sets among computing clusters by using easy programming methodologies. It is created to scale from one server to multiple numbers of machines, each of which provides local computation and storage. Rather than depending on hardware to give availability.

The library is created to recognize and handle mistakes at the application level, resulting in a very highly available service through a cluster of computers and any of them may be prone to faults or failures.Hadoop is a open source Apache framework which is written in Java language that allows distributed processing of massive data records through computer clusters via a simple programming model.Applications running in the Hadoop framework works in an surrounding that gives distributed memory and computing through computer clusters. Hadoop is scalable from one single server to hundreds or thousands machines and is designed to provide local computers and memory.

Data Analysis and Memory Storage

The complication is simple: Although storage ability of hard drives has hiked significantly with time, access speed (the rate of data that can be read from the drive) is what we are lacking. A drive from 1990 is able to store 1,370 MB of information with a transmission speed of 4.4 MB/s,4, allowing you to read every data from the drive in approximately 5 minutes. After approx. 20 years later, one TB drives are the common ones, but transmission speeds are close to 100 MB/sec, so it will take more than two and a half hours to read the whole disk. That's a long duration to be afforded on the other hand writing takes even more time. An obvious way to optimize the time is reading from multiple storage disks at the same time. Think of us having 100

drives and having 1 second of the data. Working as per the parallel approach, we will be able to read the data in not more than 120 seconds. The Ist problem to be resolved is hardware error. As you start making use of multiple parts of hardware, the probability that 1 will fail is significantly high. A usual way to get rid of data loss is via replication. Multiple copies of data are stored by the system so that the copies are available in the event of a fault. For example, RAID works this way although Hadoop's file system, the Hadoop Distributed File System (HDFS), takes a little different approach, as we will see later. The 2nd problem is that various analytical works need to be able to add data sets in some way; data read from one disk may ask for being added with data from any of the remaining 99 disks. Hadoop provides a reliable shared memory and analytics system. Memory is given by HDFS and analytics power by MapReduce. There are some other parts to Hadoop too but these capabilities are its fundamentals.

II. NEED OF HADOOP:

Big Data

We live in the age of raw facts called data. It is difficult to calculate the total variety and volume of data stored electronically, but one IDC assumption put the size of the "digital universe" at 4.4 zettabytes in 2013 and predicts a tenfold growth by 2020, to 44 zettabytes.1 One zettabyte is equal to 1021

bytes, or one thousand exabytes, one million petabytes, or one billion terabytes. That's more than one disk drive for every person in the world.

This flood of data comes from many sources.

• The New York Stock Exchange generates about 4-5 terabytes of data per day.

• There are over 240 billion photos on Facebook, a figure growing at a rate of 7 petabytes every month.

• Genealogy site Ancestry.com stores about 10 petabytes of data.

• The Internet Archive stores about 18.5 petabytes of data.

Where does Big Data come from?

The original Big Data was web data – that is, the entire Internet. Remember that Hadoop was built to index the

web. Today, Big Data comes from multiple sources.

Big Data

• Web data – this is still Big Data

• Social media data – sites like Facebook, Twitter, and LinkedIn generate a lot of data

• Clickstream data – when users browse a website, clicks are recorded for later analysis (such as

browsing patterns). Clickstream data is important in online advertising and e-commerce

• Sensor data – sensors embedded in roads to monitor traffic and other applications generate a lot of data

• Connected devices – smartphones are a great example. For example, when you use a navigation app like Google Maps or Waze, your phone sends pings that report your location and speed (this information is used to calculate traffic hotspots). Imagine hundreds of millions (or even billions) of devices consuming and generating data.

Big Data and its Benefits

In the past, only a few companies or institutions generated data that was consumed by everyone. But today, due to easy access to digital data, everyone generates and consumes it. Statistics indicate that 90% of the total data was generated in the last few years.

There are various types of data sources on offer, for example, social media data, stock market data, power grid data, transportation data, search engine data, or black box data. Big Data is really fundamental to our life and is emerging as one of the most important technologies in the modern world.

Big Data includes a wide variety of data, which is high-speed and extensible. The data in it will be of three types: 1. Structured data: relational data. 2. Semi-structured data: XML data. And 3.

Unstructured data: Word, PDF, text, multimedia records.

The data in Big Data is the type that is written once and read many times. But if this data is properly analyzed, it can help businesses make money from it. If Google can analyze your searches, it can search them in advance to get quick answers. If insurance or banking companies can analyze customer information, they can understand what they are searching for.

III. KEY FEATURES OF HADOOP:

Distributed Memory: Hadoop collects large data records through multiple machines, permitting for the memory storage and computing of massive amounts of records.

Scalability: Hadoop can increase from a single server to 1000s of machines, making it less difficult to add higher capacity as required.

Failure Tolerance: Hadoop is made to be mistake-tolerant, which means it can continue to work even in the times of hardware faults.

Data Centralization: Hadoop gives a data center technology, where data is stored and collected on a single computer where it is computed, this technology helps decrease network traffic and gives better performance.

High Availability: Hadoop gives a high availability function, which ensures that insights or information is always safe and available.

Flexible Data Management: MapReduce programming model allows distributed data management, making it less difficult to perform a variety of data management tasks.

Data Integrity: Hadoop gives an inbuilt check sum feature, which makes sure that the stored data is consistent and right.

Data Replication: Hadoop gives a data replication feature that supports replicated data across cluster for mistake tolerance.

Data Compression: Hadoop gives an inbuilt data compression function that helps decrease storage space and increase performance.

YARN: A resource manager environment that enables many data processing engines such as real time streaming, batch processing, and interactive SQL to process and modify data stored in HDFS.

IV. HADOOP ECOSYSTEM

Apache Hadoop is an open-source framework designed to handle big data challenges. It comprises several key modules:

- Hadoop Common: This module includes essential utilities that support the functionality of other Hadoop components.
- Hadoop Distributed File System (HDFS): A scalable and distributed file system that ensures high-performance access to data across applications.
- Hadoop YARN: This framework manages job scheduling and resource allocation within the cluster.
- Hadoop MapReduce: A system built on YARN that facilitates the parallel processing of extensive data sets.

In addition to these core modules, the Hadoop ecosystem incorporates various tools that enhance data access and integration with other database systems. This report primarily examines two fundamental components of Hadoop: HDFS (Hadoop Distributed File System) and YARN (Yet Another Resource Negotiator). It also explores one of the programming techniques utilized by YARN.

V. HADOOP DISTRIBUTED FILE SYSTEM (HDFS):

The Hadoop Distributed File System (HDFS) is a scalable, distributed file system designed for the Hadoop framework. Written in Java, HDFS is often seen as a data store due to its lack of POSIX compliance and inability to be mounted. It uses a single namenode and multiple datanodes, with redundancy options for the namenode. Data blocks are served over the network using a specific protocol, and communication occurs via TCP/IP sockets and remote procedure calls (RPC).

Key Features:

1. Hardware Failure: HDFS expects hardware failures and is designed for rapid, automatic recovery.
2. Streaming Data Access: Optimized for high-throughput batch processing rather than low-latency interactive use.
3. Large Data Sets: Handles files from gigabytes to terabytes, supporting high bandwidth and scalability across hundreds of nodes.
4. Simple Consistency Model: Uses a write-once, read-many approach, simplifying data consistency and enabling high-performance access.
5. Efficient Computation: Executes calculations close to the data to minimize network congestion and improve performance.
6. Portability: Easily portable across various hardware and software platforms, promoting widespread adoption.

VI. YARN ARCHITECTURE:

The fundamental idea of YARN is to split the resource management and job scheduling/monitoring functionalities into separate daemons. The idea is to have a global ResourceManager (RM) and an ApplicationMaster (AM) per application. An application is either a single job or a DAG of jobs.

The main components of the YARN architecture are.

1. Resource Manager – It primarily works as a scheduler that allocates resources to running applications.

2. Node Manager is a per-machine manager that manages and monitors the resource usage of containers and reports it to ResourceManager.

3. ApplicationMaster is a framework-specific entity. One is created per application and is not a permanent entity. It negotiates resources with ResourceManager and with the help of Node Manager manages the lifecycle of the application.

4. ResourceManager creates containers on node machines. It allocates CPU and memory. The application runs in multiple containers on the machines.

VII. CONCLUSION:

1. Scalability: As ResourceManager is completely focused on resource scheduling, more units can be

added to the cluster as the demand increases and YARN can easily manage larger clusters.

2. Multi-tenancy: Hadoop can handle larger clusters and suggest multiple organizations to share the cluster in a cost-effective manner.

3. High Cluster Utilization: As Capacity Scheduler provides capacity assurance, security, and most importantly elasticity, it maximizes the utilization of cluster resources.

4. Support for Programming Model Diversity: As Yarn split the two main functionalities in Hadoop i.e. Resource Management and Application Lifecycle, different programming models can be run on top of HDFS. Before YARN, Hadoop could process batch jobs i.e. MapReduce jobs. But YARN can now support different programming models.

VIII. REFERENCES:

[1] Tom White,Hadoop The Definitive Guide(Storage and Analysis at Internet Scale),O'Reilly 2015

[2] Dhruba Borthakur,HDFS Architecture Guide

[3] Mark Kerzner and Sujee Maniyam,Hadoop Illuminated

[4] Serge Blazhievsky and Nice Systems,Introduction to Hadoop,MapReduce and HDFS for Big Data Applications

[5] Ravi Mukkamala,Hadoop:A Software Framework for Data Intensive Computing Applications