ISSN : 2494-9150 Vol-01, Issue 11, FEB 2016.

RELIABLE STORAGE SYSTEM USING HADOOP

[1]Darshi Khatri, [2]Hetal Panchal, [3]Snehal Padge
[1,2,3]Department of IT, K J Somaiya Institute of Engineering & IT, Sion, Mumbai, Maharashtra, India.
[1]darshi.k@somaiya.edu, [2]hetal.ap@somaiya.edu, [3]snehal.padge@somaiya.edu

Abstract — **Hadoop is a quickly budding ecosystem of components based on Google's MapReduce algorithm and file system work for implementing MapReduce algorithms in a scalable fashion and distributed on commodity hardware. Hadoop enables users to store and process large volumes of data and analyze it in ways not previously possible with SQL-based approaches or less scalable solutions. Remarkable improvements in conventional compute and storage resources help make Hadoop clusters feasible for most organizations. This paper begins with the discussion of Big Data evolution and the future of Big Data based on Gartner's Hype Cycle. We have explained how Hadoop Distributed File System (HDFS) works and its architecture with suitable illustration. Hadoop's MapReduce paradigm for distributing a task across multiple nodes in Hadoop is discussed with sample data sets. The working of MapReduce and HDFS when they are put all together is discussed. Finally the paper ends with a discussion on Big Data Hadoop sample use cases which shows how enterprises can gain a competitive benefit by being early adopters of big data analytics.**

Keywords— Big data; Hadoop; Hadoop Distributed File System (HDFS); MapReduce; Replication; Fault tolerance; Unstructured data.

I. INTRODUCTION

Recent applications such as index web searches, social networking, banking transactions, recommendation engines genome manipulation in life sciences and machine learning produce huge amounts of data in the form of logs, blogs, email, and other technical structured and unstructured information streams. These data needs to be stored, processed and associated to gain close view into today's business processes. Also, the need to keep both structured and unstructured data to fulfill the government regulations in certain industry sector requires the storage, processing and analysis of large amounts of data. While a haze of excitement often envelops the universal discussions of Big Data, a clear agreement has at least combined around the definition of the term.

The term "BigData" is typically considered to be a data collection that has grown so large it can't be affordably or effectively managed using conventional data management tools such as traditional relational database management systems (RDBMS) or conventional search engines, based on the task at hand.

Another buzzing term "Big data Analytics" is where advanced analytic techniques are made to operate on big data sets. Thus, big data analytics is really about two things namely, big data and analytics and how the two have coalesced up to create one of the most philosophical trends in business intelligence (BI) today. There are several ways to store, process andanalyze large volumes of data in a massively parallel scale. Hadoop is considered as a best example for a massively parallel processing system.

A. What is hadoop?

Hadoop is an open source Apache software framework that evaluates gigabytes or petabytes of structured or unstructured data and transforms it into a more manageable form for applications to work with. As a budding technology solution, Hadoop design concerns are new to most users and not common knowledge. MapReduce

ISSN : 2494-9150 Vol-01, Issue 11, FEB 2016.

framework launched by Google by leveraging the concept of map and reduce functions are well known used in Functional Programming. Even though the Hadoop framework is written in Java language, it allows developers to deploy custom written programs coded in Java or any other language to process data in a parallel manner across thousands of commodity servers. It is optimized for adjacent read requests, whereas processing consists of scanning all the data. Based on the complexity of the process and the volume of data, response time can vary from minutes to hours. Hadoop can process the given data speedy, and it is considered as the key advantage for massive scalability. Hadoop is depicted as a solution to abundant applications in visitor behavior, image processing, web log analysis, search indexes, analyzing and indexing textual content, for research in natural language processing and machine learning, scientific applications in physics, biology and genomics and all forms of data mining. Hadoop emerged as a distributed software platform for transforming and managing large quantities of data, and has grown to be one of the most popular tools to meet many of the above mentioned needs in a cost-effective manner. By summarizing away many of the high availability (HA) and distributed programming issues, Hadoop allows developers to focus on higher level algorithms. Hence Hadoop is intended to run on a large cluster of commodity servers and to scale to hundreds or thousands of nodes.

B. Hadoop Distributed File System(HDFS)

To really understand how it is possible to scale a Hadoop cluster to hundreds and thousands of nodes, we should start with HDFS. Hadoop consist of two basic components: a distributed file system and the computational framework. In the first component of above two, data is stored in Hadoop Distributed File System (HDFS). Hadoop Distributed File System (HDFS) uses a write-once, read-many model that breaks data into blocks that it spreads across many nodes for fault tolerance and high performance. Hadoop and HDFS make use of master-slave architecture. HDFS is written in Java language, with an HDFS cluster consisting of a primary Name Node a master server that manages the file system namespace and also controls access to data by clients. There is also a Secondary Name Node which maintains a copy of the Name Node data to be used to restart the Name Node when failure occurs, although this copy may not be current and so some data loss is still likely to occur. Each Data Node manages the storage attached to the boxes that it runs on. HDFS makes use of a file system namespace that enables data to be stored in files. Each file is divided into one or more blocks, which are then shared across a set of Data Nodes. The Name Node is accountable for tasks such as opening, renaming, and closing files and data directories. The Data Node looks after block replication, creation, and removal of data when instructed to do so by the Name Node. A typical Hadoop deployment with HDFS looks like in Fig. 2.

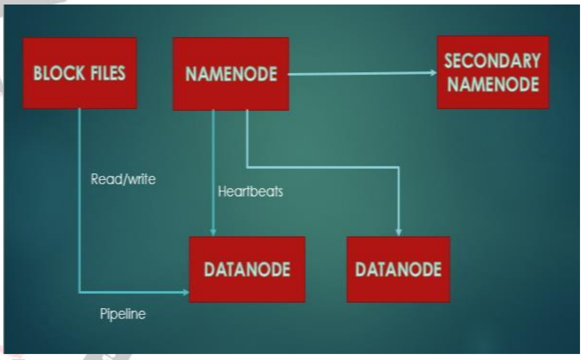

Fig 2 HDFS Architecture

C. Map Reduce Framework

Another basic component of Hadoop is MapReduce, which affords a computational framework for data processing. MapReduce is a programming replica and an associated implementation for processing and generating large data sets. MapReduce programs are inherently parallel and thus very suitable to a distributed environment. Hadoop takes a cluster of nodes to run MapReduce programs massively in parallel. A single Job Tracker schedules all the jobs on the cluster, as well as individual tasks. Here, each benchmark test is a job and runs by itself on the cluster. A job is split into a set of tasks that execute on the worker nodes. A Task Tracker running on each worker node is responsible for starting

ISSN : 2494-9150 Vol-01, Issue 11, FEB 2016.

tasks and reporting progress to the Job Tracker. As the name implies, a MapReduce program consists of two major steps, namely, the Map step processes input data and the next step Reduce assembles intermediate results into a final result. Both use key-value pairs defined by the user as input and output. This allows the output of one job to provide directly as input ’for another. MapReduce programs runs on local file system and local CPU for each cluster node. Data are broken into data blocks (usually in size of 64MB blocks), stored across the local files of different nodes, and replicated for reliability and fault tolerance. The local files constitute the file system which is called as Hadoop Distributed File System being discussed above. The number of nodes in each cluster differs from hundreds to thousands of machines. Naturally, we can write a program in MapReduce to compute the output as shown in the Fig. 3. The high-level structure would look like this:

```
mapper (filename, file-
contents): for each word in
        file-contents: emit
              (word, 1)
reducer (word,
values): sum = 0
    for each value in
    values: sum = sum +
    value
            emit (word, sum)
```

Fig 3: MR Example

II. LITERATURE SURVEYED

Over the last several years, the world has seen tremendous data growth. The requests for large scale storage have grown dramatically in science, research and business fields. Data access and I/O performance become crucial, especially in high performance computing. The size of data storage systems grows in terms of the number of storage nodes in the system. It also grows the storage capacities of individual storage nodes. Obviously, traditional file systems are insufficient to satisfy such high demand data access requests. As computers become pervasive and data size increases dramatically, data management systems’ features turn into major design issues. The features that create a problem in designing are security, scalability and availability, especially in distributed computing environments**.** A typical data management system has to deal with real-time updates by individual users, and as well as periodical large scale analytical processing, indexing, and data extraction. While such operations may take place in the same domain, the design and development of the systems have somehow evolved independently for transactional and periodical analytical processing. Such a system-level separation has resulted in problems such as data freshness as well as serious data storage redundancy**.**

Recent applications such as index web searches, social networking, banking transactions, recommendation engines genome manipulation in life sciences and machine learning produce huge amounts of data in the form of logs, blogs, email, and other technical structured and unstructured information streams. These data needs to be stored, processed and associated to gain close view into today’s business processes. Also, the need to keep both structured and unstructured data to fulfill the government regulations in certain industry sectors requires the storage, processing and analysis of large amounts of data. Various industries in the market, at present, require handling large amount of data and are thus stranded in approaching the problem with the traditional data storing system.

“Reliable storage using Hadoop” thus addressing the problem that the system faces while handling big data with Hadoop, a distributed framework, viz. a novel approach. Goal is basically to ease the handling of big data, and solve the issues arising in the security, scalability and availability of the same.

II. EXISTING SYSTEM

A. Healthcare (Storing and Processing Medical Records)

1) Problem: A health IT Company instituted a policy of saving seven years of historical claims and remit data, but

its in-house database systems had trouble meeting the data retention requirement while processing millions of claims every day.

2) Solution: A Hadoop system allows archiving seven years' claims and remit data, which requires complex processing to get into a normalized format, logging terabytes of data generated from transactional systems daily, and storing them in CDH for analytical purposes

3) Hadoop vendor: Cloudera

4) Cluster/Data size: 10+ nodes pilot; 1TB of data

day This real-time use case based on storing and processing medical records.[9]

B. Nokia

1) Problem: a)Dealing with 100TB of structureddata and 500TB+ of semi-structured data

b) 10s of PB across Nokia, 1TB / day

2) Solution: HDFS data warehouse allowsstoring all the semi/multi structured data and offers processing data at peta byte scale.

3) Hadoop Vendor: Cloudera

4) Cluster/Data size: 500TB of data & 10s of PB across Nokia, 1TB / day

Nokia collects and analyzes vast amounts of data from mobile phones. This use case was based on a case study where Nokia needed to find a technology solution that would support the collection, storage and analysis of virtually unlimited data types and volumes .[10]

C. Telecoms

1) Problem: Storing billions of mobile callrecords and providing real time access to the call records and billing information to customers. Traditional storage/database systems couldn't scale to the loads and provide a cost effective solution

2) Solution: HBase is used to store billions ofrows of call record details. 30TB of data is added monthly

3) Hadoop vendor: Intel

4) Hadoop cluster size: 100+ nodes [11]

D. Data Storage

NetApp collects diagnostic data from its storage systems deployed at customer sites. This data is used to analyze the health of NetApp systems.

1) Problem: NetApp collects over 600,000 datatransactions weekly, consisting of unstructured logs and system diagnostic information. Traditional data storage systems proved inadequate to capture and process this data.

2) Solution: A Cloudera Hadoop system capturesthe data and allows parallel processing of data.

3) Hadoop Vendor: Cloudera Cluster/Data size: 30+ nodes; 7TB of data / monthCloudera offer organizations a solution that is highly scalable with enterprise storage features that improve reliability and performance and reduce costs. [12]

E. Financial Services (Dodd-Frank Compliance at a bank)

A leading retail bank is using Cloudera and Data meer to validate data accuracy and quality to comply with regulations like Dodd-Frank

1) Problem: The previous solution using Teradataand IBM Netezza was time consuming and complex, and the data mart approach didn't provide the data completeness required for determining overall data quality.

2) Solution: A Cloudera + Data meer platformallows analyzing trillions of records which currently result in approximately one terabyte per month of reports. The results are reported through a data quality dashboard.

3) Hadoop Vendor: Cloudera + Data meer

4) Cluster/Data size: 20+ nodes; 1TB of data /month. [1]

F. Comparision :

Parameters	Existing storage system	Proposed Storage System
Type of data	Existing system can handle only Structured data.	Proposed system can handle Structured as well as Unstructured data
Speed of processing	Processing of data is low	Processing speed is high as compared to existing system.
Cost per byte	Total cost per Byte of the data is high	Total cost per Byte of data is low.
Flexibility	It is difficult to add or delete nodes, hence less flexible	It is easy to add or delete nodes, hence more flexible
Amount of data	Cannot handle large amount of data	Ability to handle large amount of data

G. Summary of Existing System

Now a days industries produces huge amounts of data in the form of logs, blogs, email, and other technical structured and unstructured information streams. It is difficult to handle this data. These data needs to be stored, processed and associated to gain close view into today's business processes. Also, the need to keep both structured and unstructured data to fulfill the government regulations in certain industry sectors requires the storage, processing and analysis of large amounts of data. Various industries in the market, at present, require handling large amount of data and are thus stranded in approaching the problem with the traditional data storing system. Proposed System thus addressing the problem that the system faces while handling big data with Hadoop, a distributed framework, viz. a novel approach. Goal is basically to ease the handling of big data, and solve the issues arising in the security, scalability and availability of the same. The issues related to data loss are dealt by forming replica of the original data in other nodes, and the formed system is fault-tolerant. We thus create a reliable system which enables addition and deletion of node at any time, so that flexibility can also be maintained.

IV. PROPOSED SYSTEM

A. Aims and Objective

The aim of this project is to make a system which basically makes the handling of big data easier and solve the issues arising in the security, scalability and availability of the data. We make use of Hadoop which supports distributed environment, so that scalability and availability of the data can be achieved.

To create a system which contains two separate data warehouses for structured and unstructured data respectively, so that any type of data handling do not lead to any difficulty, and can be handled with ease. How HDFS produces multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, exceptionally fast computations. We have implemented MapReduce concept that scales to large clusters of machines comprising thousands of machines We take a few steps in order to maintain the security of the system. One of the step is encryption and decryption using MR (Map Reduce) program. And authorization and authentication using DB (Database) layers is another step that contributes to strengthening the security of the system.

The issues related to data loss are dealt by forming replica of the original data in other nodes, and the formed system is fault-tolerant. We thus create a reliable system which enables addition and deletion of node at any time, so that flexibility can also be maintained. To make a system which basically makes the handling of big data easier and solve the issues arising in the security, scalability and availability of the data. System should be user interactive and fault tolerant. Easy interpretation of large volumes of data. Storing data in distributed storage environment. Another aim is to make a system which can store any type of data (structured or unstructured).

B. Proposed System Architecture

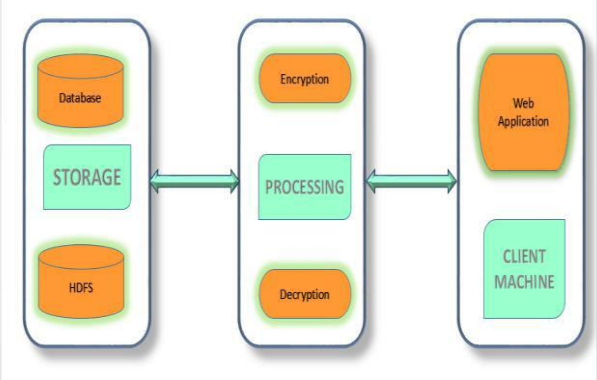

Figure 4: Proposed system architecture

In the proposed system, we have explained how Hadoop Distributed File System (HDFS) works and its architecture with suitable illustration. Hadoop's MapReduce paradigm for distributing a task across multiple nodes in Hadoop is used. The working of MapReduce and HDFS when they are put all together is proposed to use. Hadoop supports distributed environment Proposed system is three tier architecture. In which client tier contains webpage. Middle Tier contains encryption-decryption algorithms. Back Tier contains storage part.

1. Client machine: This stage is visible to the user. Webpage will be seen by the user. All the data will be entered by user through webpage. User enters id and password in the webpage. That id and password will be authenticate using database which is connected to that webpage.

2. Processing: Processing part contains MapReduce program for Encryption and Decryption. So that data which is stored in the system will be secured. When user upload the data in the system. First it will be encrypted and then stored. While downloading the data, data which is stored in system will be first decrypted and then user will get access to that data.

3. Storage: It is that part of the system where data is actually stored. This stage contains HDFS and Database. In HDFS files entered by the user will be Stored in encrypted form. Database is used for Authentication and Authorization purpose to avoid unauthorized user to enter in system.

C. Innovation done in existing system.

The proposed architecture is based on distributed data analysis through the MapReduce framework in a cloud computing environment. It is capable of storing large amount of data. it can be of any size. System is able to store that data. In order to achieve the expected data storage and processing performance. We used MapReduce framework and distributed file system. The proposed architecture enjoys following characteristics Distributed data collection from multiple sources in multiple network areas Scalable storage in a distributed file system infrastructure Scalable distributed processing in cloud environments through the MapReduce framework. Existing system can only handle structured data. Proposed system can handle structured as well as unstructured data. Proposed system will be more fault tolerant than existing system and more flexible also.

D. Proposed System Design Details

Installing VM (Virtual Machine) and setting up Hadoop environment using Hadoop 1.2.1 in fully distributed mode. Create 4 nodes Hadoop Cluster in fully distributed mode and ensure name node HA is achieved. Apache Web Server: Apache Hadoop is an excellent framework for processing, storing and analyzing large volumes of unstructured data - aka Big Data. Creation of web page: webpage will be in front end which is visible to user. Webpage includes login and creating account on drive DB layer for authentication and authorization: Database is used at the back end which contains id and password of the users. It can be used for authorization and authentication so that user identity will be verified. Hadoop HDFS reliable layer for storing the data: Back end will also contain HDFS which will be used for storing the data. It is the place where actual data will be stored. Map reduce program for encryption: Middle tier contains Map reduce program i.e. business logic layer for decryption and program triggered via web application.

ISSN : 2494-9150 Vol-01, Issue 11, FEB 2016.

Microsoft Excel: Software that allows users to organize, format, and calculate data with formulas using a spreadsheet system broken up by rows and columns. It features the ability to perform basic calculations, use graphing tools, create pivot tables and create macro programming language. Excel has the same basic features as every spreadsheet, which use a collection of cells arranged into rows and columns to organize data manipulation.

V. CONCLUSION

The data deluge -- with its three equally-challenging dimensions of variety, volume, and velocity -- has made it impossible for any single platform to meet all of an organization's data warehousing needs. Hadoop will not replace relational databases or traditional data warehouse platforms, but its superior price/performance ratio will give organizations an option to lower costs while maintaining their existing applications and reporting infrastructure. We are in the era of Big Data. Every day, we generate 2.5 quintillion bytes of data showing that the data in the world today has been created in the last two years alone. In this paper we have highlighted the evolution and rise of big data using Gartner's Hype

Cycle for emerging technologies. We have discussed how HDFS produces multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, exceptionally fast computations. We have implemented MapReduce concept that scales to large clusters of machines comprising thousands of machines. Finally the paper ends with the discussion of Real-World Hadoop use cases which helps in Business Analytics.

REFERENCES

[1] Lakshman and P. Malik, *"Cassandra: a decentralized structured storage system,"* SIGOPS Oper. Syst. Rev., vol. 44, no. 2, pp. 35–40, 2010.

[2] R. Hull and G. Zhou, *"A framework for supporting data integration using the materialized and virtual approaches,"* SIGMOD Rec., vol. 25, no. 2, pp. 481–492, 1996.

[3] Manghui Tu, Peng Li, Ling Yen, Bhavani Thuraisingham and Latifur Khan. *"Data objects replication in data grid"*. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, 5(4), 2008.

[4] Sage Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E.Long, and Carlos Maltzahn. Ceph: *"A scalable, high performance distributed file system"*. In Proc. of the 7th Conf. on Operating Systems Design and Implementation, Nov. 2006.

[5] S. A. Weil, S. A. Brandt, E. L. Miller, and C. Maltzahn. Crush: "Controlled, scalable, decentralized placement of replicated data". In Proc. of ACM/IEEE Conf. on Supercomputing, 2006.

[6] Dean, Jeffrey and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", OSDI 2004.

[7] NeilRaden,"BigDataAnalytics Architecture - Putting All Your Eggs in Three Baskets", 2012

[8] Ghemawat, Sanjay, Howard Gobioff and Shun-Tak Leung, "The Google File System" SOSP'03, Oct 19-23, ACM 2003.

[9] Cloudera Customer Case Study, "Streamlining Healthcare Connectivity with Big Data", 2012

[10] Cloudera Customer Case Study, "Nokia: Using Big Data to Bridge the Virtual & Physical Worlds", 2012.

[11] Intel Case Study, "China Mobile Guangdong Gives Subscribers Real Time Access to Billing and Call Data Records", 2012

[12] Cloudera Customer Case Study, "NetApp Improves Customer Support by Deploying Cloudera Enterprise", 2012.

[13] Cloudera Customer Case Study, "Joint Success Story: Major Retail Bank", 2012

